

# **PENGEMBANGAN SISTEM PENJUALAN DENGAN REKOMENDASI KOMPATIBILITAS KOMPONEN PADA TOKO KOMPUTER**

## **SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:

Abi Sajiwo Binar Swandito

NIM: 145150207111036



**PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA**

**MALANG**

**2021**

## PENGESAHAN

### PENGEMBANGAN SISTEM PENJUALAN DENGAN REKOMENDASI KOMPATIBILITAS KOMPONEN PATA TOKO KOMPUTER

#### SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Abi Sajiwo Binar Swandito

NIM: 145150207111036

Skrripsi ini telah diuji dan dinyatakan lulus pada

23 Juli 2021

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Achmad Arwan, S.Kom., M.Kom

NIP: 19840815 200812 1 004

Dosen Pembimbing II



Denny Sagita R., S.Kom., M.Kom

NIP: 19851124 201504 1 001

Mengetahui

Ketua Jurusan Teknik Informatika



Achmad Basuki, S.T., M.MG., Ph. D.

NIP: 19741118 200312 1 002



## PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata di dalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiaris, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 23 Juli 2021



Abi Sajiwo Binar Swandito

NIM: 145150207111036

## KATA PENGANTAR

Puji syukur penulis panjatkan kepada kehadiran Allah SWT karena berkat rahmat dan karunia-NYA, penulis dapat menyelesaikan skripsi dengan judul “PENGEMBANGAN SISTEM PENJUALAN DENGAN REKOMENDASI KOMPATIBILITAS KOMPONEN PADA TOKO KOMPUTER”. Melalui pengantar ini, penulis ingin menyatakan rasa terima kasihnya kepada berbagai pihak yang telah berkontribusi memberi dukungan baik moril maupun materil. Adapun pihak-pihak yang bersangkutan tersebut sebagai berikut:

1. Kepada orang tua penulis yang telah memberikan dukungan moril dan materiil dengan tulus kepada penulis sehingga penulis dapat menyelesaikan skripsi ini dengan baik.
2. Achmad Arwan, S.Kom., M.Kom. selaku Dosen Pembimbing Skripsi pertama, yang telah meluangkan waktu dan tenaganya untuk mengarahkan dan membimbing penulis sehingga penulis dapat menyelesaikan skripsi ini.
3. Denny Sagita Rusdianto, S.Kom., M.Kom. selaku Dosen Pembimbing Skripsi kedua, yang telah meluangkan waktu dan tenaganya untuk mengarahkan dan membimbing penulis sehingga penulis dapat menyelesaikan skripsi ini.
4. Teman-teman Teknik Informatika 2014 yang telah menghabiskan waktu bersama selama menempuh pendidikan, serta memberikan motivasi dan dorongan.

Penulis memahami bahwa penyusunan skripsi ini memiliki banyak kekurangan sehingga kritik dan saran yang membangun akan diperlukan oleh penulis. Di akhir kata, penulis juga berharap dengan keberadaannya penyusunan skripsi ini dapat membawa berkah dan manfaat bagi semua pihak yang membutuhkan dan menggunakannya.

Malang, Juli 2021

Penulis

abi.swandito@gmail.com



## ABSTRAK

**Abi Sajiwo Binar Swandito, Pengembangan Sistem Penjualan dengan Rekomendasi Kompatibilitas Komponen pada Toko Komputer.**

**Pembimbing: Achmad Arwan, S.Kom., M.Kom. dan Denny Sagita Rusdianto, S.Kom., M.Kom.**

Pada era sekarang, mempunyai sebuah komputer pribadi telah menjadi sesuatu hal yang wajar. Dahulu, komputer hanya dapat ditemukan pada gedung perkantoran atau perusahaan besar, akan tetapi, saat ini kita dapat menemukan setidaknya dalam sebuah rumah terdapat satu komputer pribadi. Kini komputer menjadi sebuah kebutuhan untuk melakukan kegiatan sehari-hari. Seiring dengan berjalannya waktu, semakin banyak peminat yang mempunyai keinginan untuk memiliki sebuah komputer pribadi. Akan tetapi, sebagian besar peminat masih belum memahami tata cara pemilihan komponen untuk membuat satu komputer pribadi seutuhnya. Dengan banyaknya jenis komponen komputer yang berada di pasaran juga menjadi salah satu hal yang membuat para peminat semakin bingung. Untuk menangani permasalahan tersebut dibuatlah pengembangan sistem penjualan dengan rekomendasi kompatibilitas komponen komputer agar peminat dapat memilih komponen komputer tanpa khawatir dengan apakah komponen komputer yang telah dipilih dan dibeli dapat bekerja dengan baik. Dalam pembuatan sistem ini menggunakan metodologi Software Development Life Cycle Waterfall. Dalam proses Implementasi perangkat lunak dilakukan dengan menggunakan *framework laravel*. Setelah sistem selesai dibuat dilakukanlah pengujian dengan menggunakan metode *white box testing* dan *black box testing*. Pada tahap *white box testing*, pengujian yang dilakukan adalah pengujian unit, pengujian integrasi, dan pengujian *black box* dengan cara melakukan pengujian validasi. Setelah melakukan tahapan pengujian dengan menggunakan metode pengujian *white box* dan pengujian *black box* didapatkan hasilnya adalah 100% berhasil.

**Kata Kunci:** Komputer pribadi, Komponen Komputer, Penjualan, Toko Komputer



## ABSTRACT

**Abi Sajiwo Binar Swandito, The Development of a Sales System with Recommendations for Compatibility of Computer Components in Computer Stores.**

**Thesis Advisor: Achmad Arwan, S.Kom., M.Kom. and Denny Sagita Rusdianto, S.Kom., M.Kom.**

In current era, owning a personal computer has become a common thing. In the past, computers could only be found in office buildings or large companies, however, nowadays we find a lot of computers in someone's home. Now computers become a necessity to do daily activities. As time goes by, a lot of people who want to own a personal computer. However, most enthusiasts still don't understand the procedure for selecting components to make a complete personal computer. In addition, there are so many types of components on the market that makes people easily confused in choosing components. To deal with this problem, the writer made the development of a sales system with recommendations for compatibility of computer components so people can choose computer components without worrying about whether the computer components that have been selected and purchased can work properly. In the making of this system using waterfall method as software development lifecycle waterfall method. In the process the software implementation is done using the laravel framework. After the systems complete, testing is carried out with white box testing and black box testing methods. White box testing and black box testing is used to test the system. After carrying out the testing stage using white box testing method and black box testing, the result are 100% succesful.

**Keywords:** Personal Computer, Computer Components, Sales, Computer Store

## DAFTAR ISI

PENGESAHAN .....	ii
PERNYATAAN ORISINALITAS .....	iii
KATA PENGANTAR .....	iv
ABSTRAK .....	v
ABSTRACT .....	vi
DAFTAR ISI .....	vii
DAFTAR TABEL .....	xii
DAFTAR GAMBAR .....	xvi
BAB 1 PENDAHULUAN .....	18
1.1 Latar belakang .....	18
1.2 Rumusan masalah .....	19
1.3 Tujuan .....	19
1.4 Manfaat .....	19
1.5 Batasan masalah .....	20
1.6 Sistematika pembahasan .....	20
BAB 2 LANDASAN KEPUSTAKAAN .....	22
2.1 Kajian Pustaka .....	22
2.2 Toko Online .....	22
2.3 Komputer .....	22
2.4 Hardware Komputer .....	23
2.4.1 Prosesor .....	23
2.4.2 Motherboard .....	23
2.4.3 Power Supply .....	24
2.4.4 Casing .....	24
2.4.5 VGA Card .....	25
2.4.6 RAM .....	25
2.4.7 Storage .....	25
2.5 SDLC Waterfall .....	25
2.6 Object Oriented Analysis dan Design .....	26



2.7	Unified Modeling Language	27
2.7.1	Use-Case Diagram	27
2.7.2	Use case Scenario	28
2.7.3	Class Diagram	29
2.7.4	Sequence Diagram	30
2.8	Entity Relationship Diagram	30
2.9	Pengujian Black Box	31
2.10	Pengujian White Box	31
2.11	HTML	32
2.12	MY SQL	32
2.13	CSS	32
2.14	PHP	32
2.15	Basis Data	32
2.16	Laravel	33
BAB 3	METODOLOGI	34
3.1	Studi Literatur	34
3.2	Analisis Kebutuhan	35
3.3	Perancangan dan Implementasi	35
3.4	Pengujian Sistem	36
3.5	Kesimpulan	36
BAB 4	ANALISIS KEBUTUHAN	37
4.1	Elisitasi Kebutuhan Sistem	37
4.2	Identifikasi Aktor	37
4.3	Kebutuhan Fungsional	38
4.4	Use Case Diagram	45
4.5	Use case scenario	45
BAB 5	PERANCANGAN DAN IMPLEMENTASI	62
5.1	Perancangan Sistem	62
5.1.1	Perancangan Sequence diagram	62
5.1.1.1	Sequence Diagram Menambah Barang Jualan	62
5.1.1.2	Sequence Diagram Mengisi Halaman Simulasi	63
5.1.1.3	Sequence Diagram Validasi Pembayaran	67



5.1.2	Perancangan <i>Class Diagram</i> .....	68
5.1.3	Perancangan Data .....	70
5.1.3.1	Perancangan <i>ERD</i> .....	70
5.1.3.2	Perancangan <i>Data Physical Data Model</i> .....	71
5.1.4	Perancangan Komponen .....	72
5.1.4.1	Perancangan Komponen <i>Class ProcessorController</i> .....	72
5.1.4.2	Perancangan Komponen <i>Class SimulasiController</i> .....	72
5.1.4.3	Perancangan Komponen <i>Class HistoryController</i> .....	73
5.1.5	Perancangan Antarmuka .....	73
5.1.5.1	Perancangan Antarmuka <i>Login User</i> .....	74
5.1.5.2	Perancangan Antarmuka <i>Login Toko</i> .....	75
5.1.5.3	Perancangan Antarmuka <i>Register User</i> .....	76
5.1.5.4	Perancangan Antarmuka Halaman Simulasi .....	78
5.1.5.5	Perancangan Antarmuka Halaman Daftar Produk .....	79
5.1.5.6	Perancangan Antarmuka Halaman Transaksi .....	81
5.1.5.7	Perancangan Antarmuka Halaman Detail Pesanan .....	82
5.1.5.8	Perancangan Antarmuka Halaman Keranjang .....	84
5.1.5.9	Perancangan Antarmuka Halaman Home Toko .....	85
5.1.5.10	Perancangan antarmuka Halaman Daftar Transaksi .....	87
5.1.5.11	Perancangan Antarmuka Halaman Detail Pesanan .....	88
5.1.5.12	Perancangan antarmuka Halaman Tambah Produk .....	90
5.1.5.13	Perancangan antarmuka Halaman Edit Produk .....	92
5.1.5.14	Perancangan antarmuka Halaman Isi Pengiriman .....	94
5.2	Implementasi Sistem .....	95
5.2.1	Implementasi Antarmuka .....	95
5.2.1.1	Implementasi Antarmuka <i>Login User</i> .....	96
5.2.1.2	Implementasi Antarmuka <i>Login Toko Komputer</i> .....	96
5.2.1.3	Implementasi Antarmuka <i>Register User</i> .....	97
5.2.1.4	Implementasi Antarmuka Halaman Simulasi .....	98
5.2.1.5	Implementasi Antarmuka Halaman Daftar Produk .....	99
5.2.1.6	Implementasi Antarmuka Halaman Transaksi .....	100
5.2.1.7	Implementasi Antarmuka Halaman Detail Transaksi .....	101



5.2.1.8	Implementasi Antarmuka Halaman Keranjang.....	102
5.2.1.9	Implementasi Antarmuka Halaman <i>Home</i> Toko Komputer..	103
5.2.1.10	Implementasi Antarmuka Halaman Daftar Transaksi.....	104
5.2.1.11	Implementasi Antarmuka Halaman Detail Transaksi .....	105
5.2.1.12	Implementasi Antarmuka Halaman Tambah Produk .....	106
5.2.1.13	Implementasi Antarmuka Halaman Edit Produk .....	107
5.2.1.14	Implementasi Antarmuka Halaman isi Pengiriman .....	108
5.2.2	Implementasi Database.....	108
5.2.3	Implementasi Kode.....	110
5.2.3.1	Implementasi Kode pada <i>Method</i> create().....	110
5.2.3.2	Implementasi Kode pada <i>Method</i> simulasi().....	111
5.2.3.3	Implementasi Kode pada <i>Method</i> bayar().....	112
BAB 6	PENGUJIAN SISTEM.....	113
6.1	Pengujian Unit.....	113
6.1.1	Pengujian Unit <i>Class</i> ProcessorController <i>Method</i> Create().....	113
6.1.2	Pengujian Unit <i>Class</i> SimulasiController <i>Method</i> Simulasi .....	116
6.1.3	Pengujian Unit <i>Class</i> History Controller <i>Method</i> Bayar .....	119
6.2	Pengujian Integrasi.....	121
6.3	Pengujian Validasi .....	125
6.3.1	Pengujian Validasi <i>User Login</i> .....	125
6.3.2	Pengujian Validasi <i>Logout</i> .....	127
6.3.3	Pengujian Validasi <i>Registrasi</i> .....	127
6.3.4	Pengujian Validasi Melihat Halaman Panduan.....	128
6.3.5	Pengujian Validasi Melihat Daftar Komponen .....	129
6.3.6	Pengujian Validasi Melihat Halaman Simulasi .....	129
6.3.7	Pengujian Validasi Pengisian Halaman Simulasi.....	130
6.3.8	Pengujian Validasi Memasukkan Barang ke Keranjang.....	132
6.3.9	Pengujian Validasi Melihat Keranjang .....	133
6.3.10	Pengujian Validasi Penghapusan Isi Keranjang .....	134
6.3.11	Pengujian Mengisi Alamat Pengiriman .....	135
6.3.12	Pengujian Validasi Pengubahan Alamat Pengiriman.....	137
6.3.13	Pengujian Validasi Pembelian Barang .....	139



6.3.14	Pengujian Validasi Melihat Daftar Pesanan .....	141
6.3.15	Pengujian Validasi Melihat Detail Pesanan .....	142
6.3.16	Pengujian Validasi Pembayaran .....	142
6.3.17	Pengujian Validasi Menerima Barang .....	143
6.3.18	Pengujian Validasi Konfirmasi Pembayaran .....	144
6.3.19	Pengujian Validasi Melihat Barang Jualan .....	145
6.3.20	Pengujian Validasi Menambah Barang Jualan .....	146
6.3.21	Pengujian Validasi Mengubah Barang Jualan .....	147
6.3.22	Pengujian Validasi Menghapus Barang Jualan .....	148
6.3.23	Pengujian Validasi Melihat Daftar Pesanan .....	148
6.3.24	Pengujian Validasi Melihat Detail Pesanan .....	149
6.3.25	Pengujian Validasi Memasukkan Data Pengiriman Barang .....	150
BAB 7	PENUTUP .....	152
7.1	Kesimpulan .....	152
7.2	Saran .....	152
DAFTAR REFERENSI .....		xviii
LAMPIRAN .....		xx

## DAFTAR TABEL

Tabel 2.1 Notasi use-case diagram .....	27
Tabel 2.2 Use case scenario .....	28
Tabel 2.3 Simbol class diagram .....	29
Tabel 2.4 Simbol sequence diagram .....	30
Tabel 4.1 Identifikasi aktor .....	38
Tabel 4.2 Kebutuhan fungsional dan spesifikasi kebutuhan .....	38
Tabel 4.3 <i>use case scenario login</i> .....	45
Tabel 4.4 <i>Use case scenario logout</i> .....	46
Tabel 4.5 <i>Use case scenario register</i> .....	46
Tabel 4.6 <i>Use case scenario</i> halaman panduan .....	47
Tabel 4.7 <i>Use case scenario</i> melihat daftar komponen .....	48
Tabel 4.8 <i>Use case scenario</i> melihat halaman simulasi .....	48
Tabel 4.9 <i>Use case scenario</i> mengisi halaman simulasi .....	49
Tabel 4.10 <i>Use case scenario</i> memasukkan barang ke keranjang .....	50
Tabel 4.11 <i>Use case scenario</i> melihat ke keranjang .....	51
Tabel 4.12 <i>Use case scenario</i> menghapus isi keranjang .....	51
Tabel 4.13 <i>Use case scenario</i> mengisi alamat pengiriman .....	52
Tabel 4.14 <i>Use case scenario</i> mengubah alamat pengiriman .....	53
Tabel 4.15 <i>Use case scenario</i> membeli barang .....	54
Tabel 4.16 <i>Use case scenario</i> melihat daftar transaksi .....	55
Tabel 4.17 <i>Use case scenario</i> melihat detail transaksi .....	55
Tabel 4.18 <i>Use case scenario</i> validasi pembayaran .....	56
Tabel 4.19 <i>Use case scenario</i> menerima barang .....	56
Tabel 4.20 <i>Use case scenario</i> konfirmasi pembayaran .....	57
Tabel 4.21 <i>Use case scenario</i> melihat barang jualan .....	58
Tabel 4.22 <i>Use case scenario</i> menambah barang jualan .....	58
Tabel 4.23 <i>Use case scenario</i> mengubah barang jualan .....	59
Tabel 4.24 <i>Use case scenario</i> menghapus barang jualan .....	59
Tabel 4.25 <i>Use case scenario</i> melihat daftar pesanan .....	60
Tabel 4.26 <i>Use case scenario</i> melihat detail pesanan .....	60



Tabel 4.27 Use case scenario memasukkan data pengiriman barang.....	61
Tabel 5.1 Perancangan komponen create(\$request) .....	72
Tabel 5.2 Perancangan komponen simulasi() .....	72
Tabel 5.3 Perancangan komponen bayar(\$id) .....	73
Tabel 5.4 Objek antarmuka <i>login user</i> .....	74
Tabel 5.5 Objek antarmuka <i>login toko</i> .....	75
Tabel 5.6 Objek antarmuka <i>register user</i> .....	77
Tabel 5.7 Objek antarmuka simulasi .....	78
Tabel 5.8 Objek antarmuka daftar produk ( <i>processor</i> ) .....	80
Tabel 5.9 Objek antarmuka transaksi .....	81
Tabel 5.10 Objek antarmuka detail pesanan <i>member</i> .....	82
Tabel 5.11 Objek antarmuka keranjang .....	84
Tabel 5.12 Objek antarmuka home toko .....	86
Tabel 5.13 Objek antarmuka daftar transaksi .....	87
Tabel 5.14 Objek antarmuka detail transaksi toko .....	89
Tabel 5.15 Objek antarmuka tambah produk ( <i>processor</i> ) .....	91
Tabel 5.16 Objek antarmuka edit produk ( <i>processor</i> ) .....	93
Tabel 5.17 Objek antarmuka isi pengiriman .....	94
Tabel 5.18 Implementasi <i>method</i> create() .....	110
Tabel 5.19 Implementasi <i>method</i> simulasi() .....	111
Tabel 5.20 Implementasi <i>method</i> bayar() .....	112
Tabel 6.1 Pseudocode <i>method</i> create() .....	113
Tabel 6.2 Kasus uji <i>method</i> menambah barang jualan .....	114
Tabel 6.3 Pseudocode <i>method</i> simulasi() .....	116
Tabel 6.4 Kasus uji mengisi halaman simulasi .....	117
Tabel 6.5 Pseudocode <i>method</i> bayar() .....	119
Tabel 6.6 Kasus uji unit bayar .....	120
Tabel 6.7 Identifikasi <i>method</i> bayar(\$id) .....	121
Tabel 6.8 Pseudocode bayar(\$id) .....	121
Tabel 6.9 Kasus uji unit <i>method</i> bayar(\$id) .....	122
Tabel 6.10 Source code kondisi foto tidak ditemukan .....	123
Tabel 6.11 Source code kondisi foto ditemukan .....	124



Tabel 6.12 Kasus uji validasi <i>login</i> .....	125
Tabel 6.13 Kasus uji validasi alternatif <i>login</i> 3a .....	126
Tabel 6.14 Kasus uji validasi alternatif <i>login</i> 3b .....	126
Tabel 6.15 Kasus uji validasi <i>logout</i> .....	127
Tabel 6.16 Kasus uji validasi register .....	127
Tabel 6.17 Kasus uji validasi alternatif register 3a .....	128
Tabel 6.18 Kasus uji validasi melihat halaman panduan .....	128
Tabel 6.19 Kasus uji validasi melihat daftar komponen .....	129
Tabel 6.20 Kasus uji validasi melihat halaman simulasi .....	129
Tabel 6.21 Kasus uji validasi alternatif melihat halaman simulasi 2a .....	129
Tabel 6.22 Kasus uji validasi mengisi halaman simulasi .....	130
Tabel 6.23 Kasus uji validasi memasukkan barang ke keranjang .....	132
Tabel 6.24 Kasus uji validasi alternatif memasukkan barang ke keranjang 1a .....	132
Tabel 6.25 Kasus uji validasi melihat keranjang .....	133
Tabel 6.26 Kasus uji validasi alternatif melihat keranjang 2a .....	133
Tabel 6.27 Kasus uji validasi penghapusan isi keranjang .....	134
Tabel 6.28 Kasus uji validasi alternatif penghapusan isi keranjang 4a .....	134
Tabel 6.29 Kasus uji validasi mengisi alamat pengiriman .....	135
Tabel 6.30 Kasus uji validasi alternatif mengisi alamat pengiriman 3a .....	135
Tabel 6.31 Kasus uji validasi alternatif mengisi alamat pengiriman 5a .....	136
Tabel 6.32 Kasus uji validasi alternatif mengisi alamat pengiriman 8a .....	136
Tabel 6.33 Kasus uji validasi pengubahan alamat pengiriman .....	137
Tabel 6.34 Kasus uji validasi alternatif pengubahan alamat pengiriman 3a .....	137
Tabel 6.35 Kasus uji validasi alternatif pengubahan alamat pengiriman 5a .....	138
Tabel 6.36 Kasus uji validasi alternatif pengubahan alamat pengiriman 8a .....	138
Tabel 6.37 Kasus uji validasi pembelian barang .....	139
Tabel 6.38 Kasus uji validasi alternatif pembelian barang 5a .....	140
Tabel 6.39 Kasus uji validasi alternatif pembelian barang 8a .....	140
Tabel 6.40 Kasus uji validasi melihat daftar pesanan .....	141
Tabel 6.41 Kasus uji validasi melihat daftar pesanan 2a .....	141
Tabel 6.42 Kasus uji validasi detail pesanan .....	142
Tabel 6.43 Kasus uji validasi alternatif melihat detail pesanan 2a .....	142



Tabel 6.44 Kasus uji validasi pembayaran.....	142
Tabel 6.45 Kasus uji validasi alternatif pembayaran 3a.....	143
Tabel 6.46 Kasus uji validasi menerima barang .....	143
Tabel 6.47 Kasus uji validasi alternatif menerima barang 3a .....	144
Tabel 6.48 Kasus uji validasi konfirmasi pembayaran .....	144
Tabel 6.49 Kasus uji validasi alternatif konfirmasi pembayaran 3a.....	144
Tabel 6.50 Kasus uji validasi melihat barang jualan.....	145
Tabel 6.51 Kasus uji validasi alternatif melihat barang jualan 3a .....	145
Tabel 6.52 Kasus uji validasi menambah barang jualan .....	146
Tabel 6.53 Kasus uji validasi alternatif menambah barang jualan 1a.....	146
Tabel 6.54 Kasus uji validasi mengubah barang jualan.....	147
Tabel 6.55 Kasus uji validasi alternatif mengubah barang jualan 3a .....	147
Tabel 6.56 Kasus uji validasi menghapus barang jualan .....	148
Tabel 6.57 Kasus uji validasi alternatif menghapus barang jualan 3a .....	148
Tabel 6.58 Kasus uji validasi melihat daftar pesanan .....	148
Tabel 6.59 Kasus uji validasi alternatif melihat daftar pesanan 2a .....	149
Tabel 6.60 Kasus uji validasi melihat detail pesanan .....	149
Tabel 6.61 Kasus uji validasi alternatif melihat detail pesanan 1a .....	149
Tabel 6.62 Kasus uji validasi memasukkan data pengiriman barang.....	150
Tabel 6.63 Kasus uji validasi alternatif memasukkan pengiriman barang 2a .....	150

## DAFTAR GAMBAR

Gambar 2.1 Hubungan relasi antar komponen komputer .....	23
Gambar 2.2 Metode SDLC Waterfall.....	25
Gambar 3.1 Diagram metodologi penelitian .....	34
Gambar 4.1 Pemodelan diagram <i>use case</i> sistem .....	45
Gambar 5.1 <i>Sequence</i> manambah barang jualan ( <i>Processor</i> ) .....	62
Gambar 5.2 <i>Sequence</i> mengisi halaman simulasi.....	64
Gambar 5.3 <i>Sequence</i> validasi pembayaran.....	67
Gambar 5.4 Rancangan umum <i>class</i> diagram.....	69
Gambar 5.5 <i>Entity Relationship Diagram</i> .....	71
Gambar 5.6 Rancangan <i>physical data model</i> .....	71
Gambar 5.7 Rancangan halaman antarmuka <i>login user</i> .....	74
Gambar 5.8 Perancangan halaman antarmuka <i>login</i> toko .....	75
Gambar 5.9 Rancangan halaman antarmuka <i>register user</i> .....	77
Gambar 5.10 Rancangan halaman antarmuka simulasi .....	78
Gambar 5.11 Rancangan halaman antarmuka daftar produk .....	80
Gambar 5.12 Rancangan halaman antarmuka transaksi .....	81
Gambar 5.13 Rancangan halaman antarmuka detail pesanan.....	82
Gambar 5.14 Rancangan halaman antarmuka keranjang .....	84
Gambar 5.15 Rancangan halaman antarmuka <i>home</i> toko .....	86
Gambar 5.16 Rancangan halaman antarmuka daftar transaksi.....	87
Gambar 5.17 Rancangan halaman antarmuka detail pesanan.....	89
Gambar 5.18 Rancangan halaman antarmuka tambah produk ( <i>Processor</i> ) .....	91
Gambar 5.19 Rancangan halaman antarmuka tambah produk ( <i>Processor</i> ) .....	93
Gambar 5.20 Rancangan halaman antarmuka isi pengiriman .....	94
Gambar 5.21 Implementasi antarmuka <i>login user</i> .....	96
Gambar 5.22 Implementasi antarmuka <i>login</i> toko komputer.....	96
Gambar 5.23 Implementasi antarmuka <i>register user</i> .....	97
Gambar 5.24 Implementasi antarmuka halaman simulasi .....	98
Gambar 5.25 Implementasi antarmuka daftar produk.....	99
Gambar 5.26 Implementasi antarmuka transaksi.....	100



Gambar 5.27 Implementasi antarmuka detail transaksi ( <i>Processor</i> ) .....	101
Gambar 5.28 Implementasi antarmuka keranjang.....	102
Gambar 5.29 Implementasi antarmuka <i>home</i> toko komputer.....	103
Gambar 5.30 Implementasi antarmuka daftar transaksi.....	104
Gambar 5.31 Implementasi antarmuka detail transaksi .....	105
Gambar 5.32 Implementasi antarmuka tambah produk ( <i>Processor</i> ).....	106
Gambar 5.33 Implementasi antarmuka edit produk ( <i>Processor</i> ) .....	107
Gambar 5.34 Implementasi antarmuka isi pengiriman .....	108
Gambar 5.35 Implementasi <i>database</i> .....	109
Gambar 6.1 Grafik <i>flow method</i> create() .....	113
Gambar 6.2 <i>Method</i> driver2() untuk uji unit <i>method</i> create() .....	116
Gambar 6.3 Grafik <i>flow method</i> simulasi() .....	117
Gambar 6.4 <i>Method</i> driver() untuk uji unit <i>method</i> simulasi().....	119
Gambar 6.5 Grafik <i>flow method</i> bayar() .....	119
Gambar 6.6 <i>Method</i> driver1() untuk uji unit <i>method</i> bayar() .....	120
Gambar 6.7 Diagram hierarki dari pengujian integrasi <i>method</i> bayar(\$id).....	121
Gambar 6.8 Grafik <i>flow method</i> bayar() .....	122
Gambar 6.9 Interpretasi hasil pengujian integrasi pertama .....	124
Gambar 6.10 Integrasi hasil pengujian integrasi kedua.....	125



## BAB 1 PENDAHULUAN

### 1.1 Latar belakang

Komputer merupakan sebuah alat yang berfungsi untuk mengolah data sesuai dengan prosedur yang telah ditentukan. Pada awalnya, istilah komputer digunakan untuk menggambarkan orang yang menggunakan atau tidak menggunakan alat bantu untuk melakukan perhitungan aritmatika, tetapi seiring berjalannya waktu arti istilah tersebut dialihkan ke mesin itu sendiri. Pada awalnya, pemrosesan informasi hampir secara eksklusif terkait dengan aritmetika, tetapi komputer modern digunakan untuk banyak tugas diluar matematika. (Suryana, 2012).

Toko komputer merupakan salah satu bentuk toko yang aktivitasnya berupa memperjualbelikan peralatan komponen komputer. Sebuah toko komputer memiliki komponen komputer yang tentunya tidak dalam jumlah sedikit serta barang tersebut tersedia dalam merek yang berbeda. Hal ini menyebabkan saat seorang pelanggan datang ke toko komputer untuk membeli sebuah komputer, ia akan meminta masukan dari si penjual untuk memastikan komponen yang akan dibeli saling kompatibel dan dapat digunakan dengan baik.

Artikel yang dibuat oleh Hale (2018) yang berjudul “31 Kesalahan Umum yang Harus Dihindari pada Saat Merakit PC” menyatakan bahwa terdapat 11 kesalahan umum yang sering dilakukan ketika memilih komponen komputer, yaitu membeli *power supply* yang cukup murah dengan kualitas rendah, mengeluarkan sejumlah uang pada komponen yang tidak penting namun mengeluarkan sedikit uang untuk komponen yang lebih penting, tidak sengaja memilih komponen komputer yang tidak saling cocok, menggunakan prosessor yang bisa di-*Overclock* dengan *motherboard* yang tidak bisa *overclock*, memilih *VGA card* yang terlalu kuat untuk performa monitor, tidak memperhatikan colokan monitor dengan *VGA card* yang dimiliki, memilih brand yang aneh untuk menghemat uang, lupa akan *Operating System* yang akan digunakan pada komputer nantinya, *casing* komputer yang menyediakan *port 3.0* akan tetapi *motherboard* tidak punya USB 3.0, dan menggunakan sejumlah uang pada komponen komputer tapi performanya tidak pernah digunakan. Hal tersebut dapat menyebabkan kerugian untuk orang-orang yang telah membeli komponen komputer tersebut tapi tidak memahami kecocokan antara komponen komputer yang satu dengan yang lainnya.

Hasil wawancara dengan Marketing Manager KiosKomputer, Fatur Rahman, adalah masih banyak orang yang tidak begitu mengerti mengenai komputer seperti DDR, *storage*, SSD, dll. Dalam sebulan, sebanyak 2-3 pelanggan kembali ke toko untuk mengembalikan barangnya karena kesalahpahaman dari pihak pelanggan mengenai hal-hal yang harus dibeli dan juga diperhatikan. Sehingga, pada saat barang yang dibeli tersebut sampai ke rumah dan dicoba di komputer rumah ternyata barang tersebut tidak sesuai dengan komponen komputer yang ada di rumah sehingga tidak dapat digunakan. Masih banyak orang yang ingin membeli suatu prosessor versi yang bisa di-*overclock*, tetapi menggunakan



*motherboard* yang tidak bisa untuk *overclock* sehingga pihak toko harus mengarahkan konsumen agar pelanggan tidak menyesal nantinya. Hal ini akan memakan banyak waktu untuk mengarahkan pelanggan dibandingkan dengan waktu transaksi untuk jual beli. Pelanggan berisiko untuk mengalami kerugian jika terkena penipuan bila tidak membeli komponen komputer di tempat yang bisa dipercaya serta bila ada kejadian kesalahan membeli komponen komputer akan menghabiskan waktu untuk mengembalikan barang tersebut ke toko serta bila mana toko tersebut tidak menerima pengembalian barang karena kesalahan pembeli maka akan membuat rugi pihak pembeli dari segi keuangan.

Berdasarkan permasalahan yang telah diuraikan oleh penulis di atas, diperlukannya sebuah cara untuk pembeli dapat memilih komponen komputer yang diinginkan tanpa harus khawatir dengan barang yang akan dibeli tidak dapat berkerja dengan baik. Sehingga dibutuhkan sebuah sistem yang tidak hanya menjual komponen komputer tapi juga memandu penggunaanya dalam memilih komponen komputer.

## 1.2 Rumusan masalah

Berdasarkan latar belakang di atas, dapat dirumuskan masalah sebagai berikut:

1. Bagaimana hasil analisis dan pembangunan sistem penjualan dengan rekomendasi kompatibilitas komponen pada toko komputer?
2. Bagaimana hasil implementasi dari sistem penjualan dengan rekomendasi kompatibilitas komponen pada toko komputer ?
3. Bagaimana hasil pengujian dari sistem penjualan dengan rekomendasi kompatibilitas komponen pada toko komputer?

## 1.3 Tujuan

Tujuan dari penelitian ini adalah untuk mengembangkan sistem penjualan dengan rekomendasi kompatibilitas komponen pada toko komputer:

1. Mengembangkan sebuah sistem penjualan dengan rekomendasi kompatibilitas komponen pada toko komputer.
2. Melakukan perancangan dan mengimplementasikannya pada sistem penjualan dengan rekomendasi kompatibilitas komponen pada toko komputer berdasarkan analisis kebutuhan yang telah dilakukan sebelumnya.
3. Melakukan pengujian kesesuaian kebutuhan yang telah dilakukan sebelumnya.

## 1.4 Manfaat

Manfaat yang diharapkan penulis dari pelaksanaan penelitian ini adalah :

1. Sistem ini diharapkan dapat menuntun pengguna dalam pembelian komponen komputer.



2. Hasil penelitian ini diharapkan dapat menjadi referensi yang berguna dan bermanfaat untuk penelitian - penelitian yang akan datang.

### 1.5 Batasan masalah

Permasalahan pada penelitian ini dibatasi pada fokus tertentu guna mencapai tujuan yang diharapkan. Adapun batasan masalah yang dikaji sebagai berikut:

1. Dalam pengembangan sistem penjualan dengan rekomendasi kompatibilitas komponen pada toko komputer platform yang digunakan pada sistem berbasis web.
2. Penambahan data pada sistem hanya dapat dilakukan oleh satu toko komputer yang terkait.
3. Pembayaran pada sistem penjualan dengan rekomendasi kompatibilitas komponen pada toko komputer dilakukan langsung ke rekening toko komputer.

### 1.6 Sistematika pembahasan

Sistematika dari penulisan laporan berisi tentang metode penulisan yang digunakan untuk membuat laporan tugas akhir. Berikut uraian sistematika penulisan:

#### BAB I PENDAHULUAN

Bab ini menjelaskan gambaran umum mengenai latar belakang penelitian, perumusan masalah penelitian, pembatasan masalah penelitian, tujuan dilakukannya penelitian, manfaat dari penelitian, serta sistematika penulisan.

#### BAB II LANDASAN KEPUSTAKAAN

Bab ini menjelaskan tentang landasan teori yang berhubungan dan berkaitan dengan tema penulisan untuk dijadikan pedoman utama dalam perancangan sistem penelitian ini. Hal tersebut meliputi tentang komponen komputer, HTML, My SQL, CSS, PHP, basis data, UML, *usecase diagram*, *sequence diagram*, *class diagram*.

#### BAB III METODELOGI PENELITIAN

Bab ini menjelaskan tentang tahapan – tahapan yang dilakukan dalam waktu penelitian yang terdiri dari studi literatur, metode pengambilan data, analisis kebutuhan, perancangan sistem, implementasi sistem, pengujian sistem, dan penutup.

#### BAB IV ANALISIS KEBUTUHAN

Bab ini menjelaskan tentang cara mendapatkan elisitasi kebutuhan, identifikasi aktor sistem, kebutuhan fungsional sistem, pemodelan *use case diagram*, dan *use case scenario*.



## BAB V PERANCANGAN DAN IMPLEMENTASI

Bab ini menjelaskan tentang pemodelan *sequence diagram*, *class diagram*, *entity diagram*, perancangan antarmuka sistem, perancangan komponen, implementasi antarmuka, implementasi data, dan implementasi *code*.

## BAB VI PENGUJIAN SISTEM

Bab ini menjelaskan tentang bagaimana pengujian sistem berlangsung dan bagian apa saja yang diujikan.

## BAB VII PENUTUP

Bab ini berisikan kesimpulan yang didapatkan dari bab – bab sebelumnya dan rekomendasi atas hasil yang diperoleh, dan harapan untuk dapat membantu pengembangan di masa mendatang.



## BAB 2 LANDASAN KEPUSTAKAAN

### 2.1 Kajian Pustaka

Cara menghindari kesalahan umum pada saat merakit komputer adalah dengan mengetahui bahwa tidak semua komponen komputer saling bersesuaian satu sama lain. Sebagai gambaran, terdapat hubungan antara *processor* dengan *Motherboard*. *Processor* intel generasi kedua dan ketiga membutuhkan *Motherboard* yang memiliki soket LGA 1151, namun untuk *processor* generasi keempat walaupun dari brand yang sama, yaitu intel, tetapi *Motherboard* yang digunakan berbeda, yakni *Motherboard* yang bersoket LGA 1150. (Chachos, 2013)

Pada penelitian yang dilakukan oleh Mustofa (2015) dengan judul penelitian Perancangan *E-Commerce* Penjualan Komputer dan Alat Elektronik Berbasis Web Pada Toko Damar Komputer Pringsewu dilakukan dengan tujuan sistem yang dibuat dapat menjangkau pelanggan pada Toko Damar Komputer yang berlokasi jauh dari lokasi toko. Pada penelitian ini sistem yang dibuat memiliki 4 fungsi utama yaitu melakukan pemesanan, melihat pesanan, melihat produk, melihat keranjang belanja.

Keputusan adalah proses memilih antar dua atau lebih pilihan yang tersedia. Dengan kata lain, harus ada pilihan alternatif. Di sisi lain, jika konsumen tidak memiliki pilihan, mereka tidak dapat diklasifikasikan sebagai pengambilan keputusan. Tidak semua konsumen membutuhkan informasi yang sama pada tingkat pencarian untuk mengambil keputusan. Pengambilan keputusan memerlukan proses keputusan. (Mujiyana, 2013)

Dari penelitian sebelumnya, peneliti tidak dapat menemukan penelitian dengan judul yang sama dengan judul penelitian penulis, namun penulis menyimpan beberapa penelitian yang terkait untuk referensi yang bertujuan untuk memperkaya bahan kajian dalam penelitian penulis.

### 2.2 Toko Online

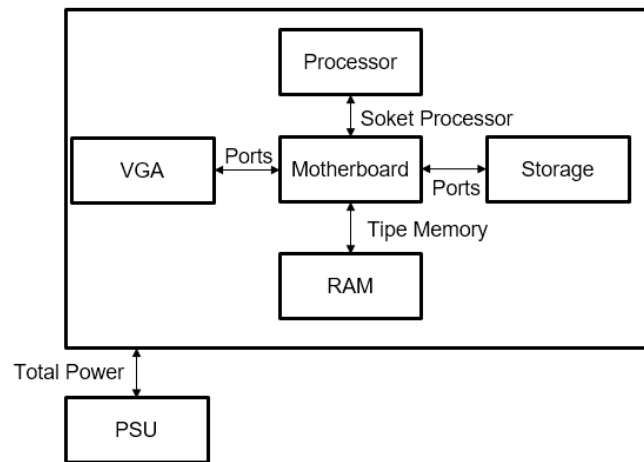
Toko *online* terdiri dari dua suku kata berupa toko dan *online*. Berdasarkan Kamus Besar Bahasa Indonesia (KBBI), kata toko merupakan tempat atau bangunan permanen yang digunakan penjual untuk menjual barangnya. Sedangkan, *online* merupakan sebuah status seseorang sedang terhubung dalam sebuah jaringan. Berdasarkan pengertian tersebut, disimpulkan bahwa toko online merupakan tempat terjadinya aktivitas perdagangan yang saling terhubung dalam sebuah jaringan (Sarwandi, 2016).

### 2.3 Komputer

Komputer adalah sebuah perangkat elektronik yang berkemampuan untuk menerima dan memproses masukkan dari pengguna. Kemudian, komputer dapat menampilkan informasi dengan menjalankan sebuah program yang ada pada memori komputer (Fauzi, 2019). Komputer tersusun dari beberapa komponen



yang dapat terhubung bekerja satu dengan yang lainnya dengan baik. Jika sebuah komponen komputer tidak dapat bekerja sama dengan komponen lain, komputer tersebut pasti tidak akan berjalan dengan baik. Adapun relasi hubungan antara komponen komputer disajikan pada gambar 2.1.



Gambar 2.1 Hubungan relasi antar komponen komputer

## 2.4 Hardware Komputer

Perangkat keras komputer (*hardware*) merupakan komponen fisik yang dapat diraba oleh manusia, komponen – komponen komputer bersatu untuk membentuk sebuah sistem komputer yang dapat berkerja dengan baik. Umumnya komponen – komponen ini diletakkan ke dalam sebuah *casing* komputer dan terdapat beberapa komponen lainnya di luar dari casing komputer (Sutono, 2014). Perangkat-perangkat yang berada di dalam casing komputer umumnya adalah *processor*, *motherboard*, *power supply*, *Casing*, *RAM*, *VGA card*, dan *storage*.

### 2.4.1 Prosesor

*Processor* merupakan sebuah komponen yang penting dalam sebuah komputer, tanpa komponen satu ini komputer tidak dapat berjalan atau tidak akan berguna sama sekali hal ini dikarena *processor* merupakan otak dari sebuah komputer. Sedikit membahas sejarah *processor*, perkembangan *processor* dimulai oleh *processor* yang di hadirkan dari sebuah perusahaan bernama *intel*. Pada masa itu merupakan satu satunya *microprocessor* yang ada di bumi ini. Namun, pada era ini kita dapat banyak melihat *processor* telah beredar dari produsen yang lain (Mufadhol, 2017).

### 2.4.2 Motherboard

*Motherboard* memiliki tugas sebagai tempat yang menyediakan wadah untuk meletakkan berbagai macam komponen komputer. Seperti, memori, *BIOS*, *processor*, dan lain - lain. Pada *motherboard* terdapat banyak slot slot *socket* sebagai tempat untuk meletakkan komponen - komponen komputer. Seperti, slot



soket *processor* yaitu sebuah tempat yang disediakan untuk meletakkan *processor* dengan karakteristiknya yang sesuai dengan tipe *socket* yang ada pada *processor*.

Slot ekspansi pada *motherboard* menjadi sebuah tempat untuk meletakkan berbagai jenis komponen, yaitu *Modem*, *VGA card*, *sound card*, *MPEG Card*, dan *LAN card*. Slot ekspansi biasanya berbentuk persegi panjang yang terdiri dari 8 – 128 bit. Slot PCI atau biasa disebut dengan saluran lokal bus yang terdiri dari 32 bit dan 64 bit yang mengikat bus ekspansi secara langsung ke *processor*. Umumnya slot ini menjadi tempat untuk meletakkan *video capture card*, *sound card*, *VGA card*, dan *modem card*. Khusus untuk slot *VGA* menggunakan desain baru dengan menggunakan saluran 128 bit hal ini dilakukan dengan tujuan peningkatan kinerja video adapter 3D yang biasa disebut *AGP*. Slot *memory/RAM* adalah sebuah slot untuk meletakkan *RAM* (Yani, 2005).

Pada bagian *motherboard* terdapat sebuah *chipset* yang berguna untuk mengatur fungsi – fungsi yang ada. Jenis *chipset* dari *motherboard* dapat menentukan tipe *processor* mana saja yang kompatibel dengan *motherboard*. Selain itu, jenis *chipset* juga menentukan jenis memori yang dapat mendukung sistem PC dan beserta berapa kapasitas maksimum *memory ram*, kelengkapan I/O yang mampu disediakan, dll. *USB* merupakan salah satu *port* komunikasi dari *motherboard* terhadap eksternal *memory* seperti *hardisc* eksternal atau *USB flash drive*. Konektor *power* yang berada pada *motherboard* berguna untuk menghubungkan *motherboard* dengan *power supply*. Konektor *power* pada *motherboard* memiliki jenis yang berbeda sesuai dengan tipe *motherboard*nya (Yani, 2005).

### 2.4.3 Power Supply

Sebuah komputer pasti akan memiliki sebuah *power supply* yang berfungsi untuk mengubah tegangan arus listrik AC menjadi arus DC sehingga komponen komputer yang membutuhkan tegangan listrik berarus DC mendapatkan tegangan listrik, seperti *motherboard*, *hardisk*, dan kipas. Selain membutuhkan arus listrik, komputer juga membutuhkan peran *power supply* untuk dapat menyala dengan baik (Mufadhol, 2017).

*Power supply* pada umumnya berbentuk kotak yang dapat ditemukan pada bagian belakang *casing* komputer. Besarnya kapasitas arus listrik yang dapat ditangani oleh *power supply* dihitung dengan satuan watt misalnya 500 watt. *Power supply* memiliki banyak jenis kabel konektor yang memiliki fungsi yang berbeda beda pula (Mufadhol, 2017).

### 2.4.4 Casing

*Casing* menjadi sebuah tempat yang melindungi komponen komputer terhadap gangguan eksternal. Terdapat 2 jenis *casing*, yaitu *desktop* dan *tower*. *Desktop* merupakan komputer yang sudah dirakit dari pabriknya atau biasa disebut dengan komputer *build up*. Sedangkan, *tower* adalah komputer yang harus dirakit dulu (Yani, 2005).



#### 2.4.5 VGA Card

Seiring berjalannya waktu pemrosesan grafis yang semula yang hanya mengandalkan kerja *processor* telah berubah dengan adanya *VGA Card* yang dapat merendering grafis 3D secara penuh dan efisien. Setelah adanya *VGA card* beban utama dari *processor* menjadi lebih ringan karena proses pengolahan grafis dialihkan kepada *VGA Card* yang memiliki GPU (*Graphic Processing Unit*). Hal yang harus diperhatikan dalam pemasangan *VGA Card* ke *motherboard* adalah dengan memperhatikan jenis kartu grafis dengan slot *motherboard*. (Yani, 2005).

#### 2.4.6 RAM

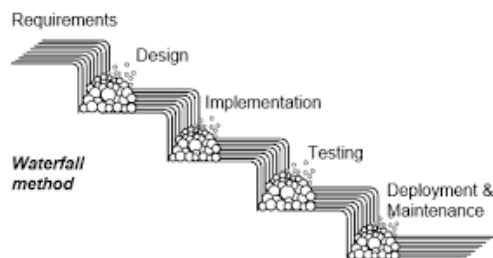
*Random Access Memory* atau yang biasa disebut dengan *RAM* merupakan *memory* eksternal untuk penyimpanan program dan data yang dapat diakses oleh *processor*. *RAM* merupakan sebuah *memory* namun dengan sifat yang sementara selama aliran listrik tidak terputus dari komputer. Bila mana komputer tidak memiliki arus komputer maka data yang tersimpan pada *RAM* akan hilang (Yani, 2005).

#### 2.4.7 Storage

*Storage* atau *Hardisk* merupakan suatu perangkat penyimpanan sekunder. *Harddisk* dibuat untuk dapat menyimpan data dalam kapasitas yang besar hal ini bertujuan untuk menampung program aplikasi yang tidak memungkinkan tersimpan dalam 1 disket, serta dibutuhkan sebuah media yang dapat menyimpan berkas dalam jumlah besar hal ini dapat dijumpai pada media penyimpanan *database* pada suatu perusahaan (Mufadhol, 2017).

### 2.5 SDLC Waterfall

Pembangunan sistem ini menggunakan model *SDLC waterfall*. Terdapat lima tahapan dalam mode *SDLC waterfall*, yaitu tahapan *requirement*, *design*, *implementasion*, *testing*, *deployment*, dan *maintenance* yang tersajikan pada Gambar 2.1.



Gambar 2.2 Metode SDLC Waterfall

Sumber : Marsic (2012)

Menurut Marsic Ivan (2012) dalam *SDLC waterfall* terdapat beberapa tahapan. Berikut tahapan pada model *waterfall* :



### 1. *Requirements*

Pada tahapan ini, dilakukan sebuah penggalan kebutuhan secara intensif, yang berguna untuk menspesifikasikan kebutuhan, menentukan apa saja fitur perangkat lunak yang akan dibuat, dan menentukan tujuan sistem perangkat lunak yang akan dibuat sebagai tahap awal.

### 2. *Design*

Pada tahapan ini, dilakukan sebuah aktivitas untuk mendesain sebuah sistem berdasarkan dari hasil penggalan kebutuhan. Hal ini dilakukan untuk menjadikan desain sebagai dasar implementasi pembangunan perangkat lunak.

### 3. *Implementation*

Pada tahap implementasi, dikerjakan dengan melakukan pengetikan kode program sesuai dengan yang telah di desain pada fase sebelumnya. Tahapan ini dimulai dari pembuatan unit - unit fungsi hingga pengintegrasian menjadi sebuah sistem yang utuh sesuai dengan kebutuhan yang telah didefinisikan.

### 4. *Testing*

Pada tahap ini, perangkat lunak telah diuji untuk dipastikan bahwa sistem yang dibuat dapat menjawab kebutuhan sistem yang telah didefinisikan pada tahap awal. Hal ini dilakukan untuk mengurangi kesalahan pada sistem.

### 5. *Deployment and Maintenance*

Pada tahap ini, terbagi menjadi 2 tahapan yaitu tahapan *deployment* dan *maintenance*. Pada tahapan *deployment*, akan diketahui bagaimana kualitas dari perangkat lunak serta masukkan dari pengguna. Pada tahapan *maintenance*, dilakukan setelah perangkat lunak yang dibuat di gunakan oleh pengguna. Proses ini dapat terjadi bila mana tidak ada kesalahan saat melakukan pengujian pada sistem.

## 2.6 *Object Oriented Analysis dan Design*

Pengertian *Object Oriented Analysis dan Design with the Unified Process* atau yang biasa disingkat menjadi OOAD menurut Satzinger Et al. (2010) dapat dilihat dibawah ini :

- Object Oriented Analysis* atau yang biasa disingkat dengan OOA merupakan segala jenis objek yang dapat melakukan sebuah aksi dan menunjukkan sebuah interaksi kepada pengguna sistem untuk dapat menyelesaikan tugas – tugas yang ada pada sistem. Di dalam sistem komputer, objek adalah sesuatu hal yang dapat merespon pesan.
- Object Oriented Design* atau yang biasa disingkat dengan OOD merupakan berbagai jenis macam objek yang melakukan interaksi



dengan pengguna atau sebuah perangkat yang berada pada sistem. Menggambarkan bagaimana aksi objek dalam menyelesaikan tugas, serta dilakukan penyempurnaan definisi dari tiap – tiap objek sehingga dapat diterapkan dengan bahasa tertentu.

- c. *Object Oriented Programming* atau yang biasa disingkat dengan OOP merupakan pernyataan untuk mendefinisikan tipe objek dalam bentuk bahasa pemrograman.

## 2.7 Unified Modeling Language

*Unified Modeling Language* atau yang biasa disingkat dengan (UML) adalah sebuah bahasa dalam pemodelan untuk menggambarkan sebuah sistem berdasarkan objek – objek yang ada di sistem. UML hanya dapat menentukan notasi standar yang digunakan dalam permodelan objek dalam suatu pengembangan sistem (Whitten, 2007).

Di bawah ini adalah empat macam dari UML yang biasa digunakan pada pengembangan sebuah sistem :


### 2.7.1 Use-Case Diagram


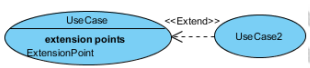
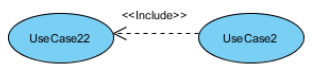

Diagram *Use-Case* merupakan diagram yang menunjukkan interaksi antara pengguna dengan sistem internal (Whitten, 2007). Secara grafis, diagram ini mendeskripsikan siapa yang menggunakan sistem dan menunjukkan ekspektasi pengguna ketika berinteraksi dengan sistem tersebut. Adapun unsur-unsur yang harus terpenuhi sebagai berikut:

1. *Use Cases* merupakan sebuah kumpulan fungsi yang ada pada sistem, di mana fungsi tersebut dapat di akses oleh pengguna untuk melakukan sebuah pekerjaan.
2. *Actors* merupakan pihak yang melakukan interaksi dengan sistem untuk pertukaran informasi baik dari pihak pengguna maupun dari luar sistem.
3. *Relationships* merupakan garis penghubung antara pengguna dengan *use cases*.

Dalam pembuatan use case diagram terdapat beberapa notasi yang dapat dilihat pada tabel 2.1.

Tabel 2.1 Notasi use-case diagram

Nama Notasi	Keterangan	Simbol
<i>Use Case</i>	<i>Use case</i> menunjukkan gambaran ketika pengguna akan menggunakan sistem.	

Aktor	Aktor merupakan pihak yang menggunakan sistem. Aktor tidak hanya merupakan seorang manusia melainkan dapat juga sebuah aplikasi yang memberikan sebuah <i>input</i> ke dalam sistem.	 Actor
Include	Sebuah <i>use case</i> dapat menggunakan fungsionalitas yang terdapat pada <i>use case</i> lainnya.	
Extend	Mendefinisikan bahwa <i>use case</i> tersebut memperluas perilaku dari <i>use case</i> yang lainnya.	
Association	Digunakan untuk menghubungkan aktor dengan sebuah <i>use case</i> .	

Sumber : Whitten & Bentley (2007)

### 2.7.2 Use case Scenario

*Use case scenario* merupakan sebuah narasi dokumen yang menjelaskan fungsi-fungsi yang dibutuhkan dalam sebuah *use case* secara rinci. *Use case descriptions* menjelaskan tujuan akhir dari *use case* serta memberikan gambaran tentang peristiwa yang biasanya terjadi. Sehingga, penulisan deskripsi diharuskan serinci mungkin baik urutan kejadian maupun *scenario* yang berhubungan dengan *use case*. Adapun contoh dari *use case scenario* tersajikan pada Tabel 2.2 sebagai berikut.

Tabel 2.2 Use case scenario

Use Case	Login	
Actor	User	
Description	Use case ini menjelaskan tentang proses <i>login</i> ke dalam akun	
Precondition	User telah terdaftar.	
Flow of Events	User Action	System Response
	1. Memasukkan <i>username</i> dan <i>password</i> .	2. Memvalidasi apakah <i>username</i> telah terdaftar dan apakah <i>password</i> yang dimasukkan benar atau salah.
		3. Menampilkan halaman utama sistem.



Post Condition

User masuk ke halaman utama.

Sumber : Britton & Doake (2005)

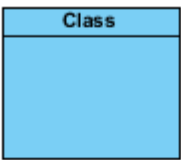



### 2.7.3 Class Diagram

*Class diagram* merupakan sebuah pemodelan terstruktur dari ojek yang ada di dalam sistem, yakni memperlihatkan *object class* berserta hubungan antara tiap *object class* (Whitten, 2007). Dalam pemodelan *class diagram* terdapat beberapa tahapan, di antaranya adalah :

- Mengidentifikasi hubungan asosiasi dan keberagaman dari kelas yang ada dari objek. pada tahapan ini, mengidentifikasi hubungan yang ada dari kelas objek yang ada, berupa informasi apa yang dibutuhkan dari tiap objek.
- Mengidentifikasi hubungan *general* dan relasi khusus antar kelas. Mencari tahu hubungan antar kelas yang ada termasuk ke hubungan umum atau hubungan khusus. Yang dimaksud adalah pengklasifikasian sebuah hierarki, sebuah hubungan berdasarkan *class abstract*, *parent*, dan *child class*.
- Mengidentifikasi hubungan agregasi. Pada bagian ini, menentukan bagaimana hubungan agregasi yang terjadi antar *class* yang ada pada sistem, yang berarti hubungan satu objek merupakan bagian dari objek lain.
- Mempersiapkan *class diagram*. Pada bagian ini membuat pemodelan *class diagram* berdasarkan dari informasi hubungan antar kelas yang ada di sistem, maupun hubungan agregasi antar *class*.

Berikut merupakan beberapa simbol yang ada pada pemodelan *class diagram* disajikan pada tabel 2.3.

Tabel 2.3 Simbol class diagram



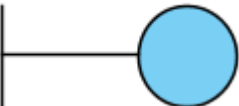


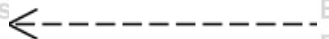
Nama Komponen	Simbol	Keterangan
Class		Deskripsi dari objek yang dibagi menjadi 3 bagian, yaitu nama kelas, <i>attribute</i> , dan <i>operation</i> .
Aggregation		Bentuk dari hubungan asosiasi dengan ketentuan spesifik.
Association		Bentuk dari hubungan terstruktur antar <i>class</i> .
Generalization		Bentuk dari hubungan yang lebih <i>general</i> atau lebih spesifik.

Sumber : Whitten & Bentley (2007)

#### 2.7.4 Sequence Diagram

Sequence diagram merupakan sebuah diagram pemodelan yang menampilkan interaksi antar aktor dan sistem dalam sebuah skenario. Pada tahap ini, kita fokus memikirkan keseluruhan sistem yang ada, tidak dulu menganalisa lebih lanjut mengenai *individual object class*. Diagram ini menggambarkan alur keluar masuk data pada sistem. Pada *sequence diagram* memungkinkan sebuah use case dapat memiliki lebih dari satu *sequence diagram* dalam pemodelan keseluruhan *use case* tersebut. Terdapat beberapa simbol dalam pembuatan *sequence diagram*, penjelasan mengenai simbol pada *sequence diagram* disajikan pada tabel 2.4.

Tabel 2.4 Simbol *sequence diagram*

Nama Komponen	Simbol	Keterangan
Aktor	 Actor	Pengguna yang terlibat dalam suatu sistem.
Entity		Menggambarkan sebuah <i>database</i> atau model.
Boundary		Menggambarkan sebuah dinding pemisah antar pengguna dengan sistem, biasanya sebuah halaman atau <i>form</i> .
Controller		Menggambarkan hubungan di antara <i>boundary</i> dan <i>entity</i> .
Message		Menunjukkan arah tujuan antar objek.
Message (return)		Menunjukkan arah kembali antar objek.

Sumber : Whitten & Bentley (2007)

#### 2.8 Entity Relationship Diagram

*Entity Relationship Diagram* atau yang biasa disingkat dengan ERD adalah permodelan data yang menggunakan beberapa notasi untuk menggambarkan sebuah data yang berada pada konteks entitas, serta menggambarkan hubungan pada data tersebut (Whitten, 2007). Pemodelan ERD digunakan dengan tujuan



untuk menggambarkan logika struktur pada *database* dalam bentuk diagram, dan juga menyajikan desain *database* dengan sederhana dan mudah di pahami (Connolly, 2010).

## 2.9 Pengujian *Black Box*

Pengujian *black box* merupakan pengujian yang dilakukan terhadap syarat fungsional sistem, sehingga *developer* sistem mendapatkan prosedur masukan yang akan sepenuhnya melaksanakan persyaratan fungsional dalam sebuah sistem. Pada tahapan *black box* testing kesalahan yang dicari adalah sebagai berikut :

1. Fungsi yang tidak valid
2. Kesalahan perilaku.
3. fungsi yang hilang.
4. Kesalahan kinerja.
5. Kesalahan dalam inisialisasi dan pemutusan.
6. Kesalahan antarmuka.

## 2.10 Pengujian *White Box*

Pengujian *white box* adalah sebuah metode desain uji kasus untuk menghasilkan uji kasus dengan menggunakan struktur kontrol desain procedural. Dalam penggunaan metode pengujian *white box*, diperoleh uji kasus yang :

1. Memberikan jaminan setidaknya semua jalur independen telah digunakan setidaknya satu kali.
2. Memberikan jaminan semua keputusan logis dari sisi *true* dan *false*.
3. Memberikan jaminan bahwa semua batas fungsi perulangan dan batas operasionalnya telah di jalankan.
4. Memberikan jaminan validitasnya struktur internal.

Dalam melakukan proses pengujian dibuat terlebih dahulu *test case* dengan penerjemahan *flowchart* kedalam notasi *flowgraph*. Terdapat beberapa istilah dalam pembuatan *flowgraph* yang dapat dilihat di bawah ini:

1. *Node* merupakan sebuah lingkaran pada *flowgraph* yang menggambarkan satu atau lebih perintah.
2. *Edge* merupakan sebuah tanda panah yang menggambarkan alur dari setiap node yang memiliki tujuan ke *node* yang lainnya.
3. *Region* merupakan sebuah daerah yang dibatasi oleh *node* dan *edge*. Untuk perhitungan daerah diluar *flowgraph* harus juga dihitung.
4. *Predicate Note* adalah sebuah kondisi yang ada pada node dan memiliki karakteristik dua atau lebih *edge*.
5. Cyclomatic Complexity adalah sebuah metrik dari sistem yang menyediakan ukuran yang dapat dihitung dari tingkat kompleksitas alur



logikal program dan dapat digunakan dalam pencarian jumlah *path* yang ada.

6. *Independen Path* merupakan sebuah jalur lintas atau yang akan melalui sistem di mana setidaknya terdapat proses perintah yang baru atau kondisi yang baru.

## 2.11 HTML

*Hyper Text Markup Language* atau yang biasa disingkat dengan HTML adalah sebuah file yang berisikan sekumpulan simbol – simbol atau tag - tak. HTML dibuat untuk dapat menampilkan halaman pada *web browser*. Tag-tag pada HTML selalu diawali dengan <x> dan diakhiri dengan </x> dimana x tag HTML seperti *body* , *div*, *table*, *b*, dll. (Anhar, 2010)

## 2.12 MY SQL

MySQL merupakan sebuah *tools* dalam manage sistem basis data atau biasa dikenal dengan RDBMS (*Relational Database Management System*), MySQL AB dapat menjadikan MySQL menjadi sebuah perangkat lunak gratis yang dapat digunakan oleh berbagai pihak dalam lisensi dari *GNU General Public License* (GPL), akan tetapi tetap menjual versi khusus di bawah lisensi komersial. Lisensi ini bebas digunakan oleh semua orang, akan tetapi tidak diperbolehkan dijadikan sebuah produk turunan yang bersifat komersial. Dengan menggunakan SQL kita dapat dengan mudah melakukan pemilihan atau seleksi dan pemasukan data.

## 2.13 CSS

*Cascading Style Sheet* atau yang biasa disingkat CSS merupakan sebuah bahasa pemrograman yang digunakan untuk format HTML agar penampilannya menjadi lebih cantik dan efektif. Jika kita ingin membuat website yang dapat digunakan oleh beberapa *device* maka gunakanlah yang sering disebut dengan *responsive* dan dapat dipelajari dalam CSS (Yudhanto, 2018).

## 2.14 PHP

Hypertext Preprocessor atau yang biasa disingkat menjadi PHP merupakan sebuah kode script pada sisi server, biasanya diletakan pada bagian HTML. PHP merupakan sebuah server-side embedded script language yang berarti code di jalankan dari server dan disampaikan ke web browser, akan tetapi proses dilakukan secara keseluruhan di server (Peranginangin, 2006).

## 2.15 Basis Data

Basis data merupakan sebuah kumpulan data yang terkait guna memenuhi kebutuhan informasi pada sebuah organisasi. Kumpulan data tersebut dapat berasal dari sistem yang berinteraksi dengan basis data secara bersamaan dengan *Database Management system* (DBMS) dan basis data itu sendiri (Connolly, 2010).



Basis data merupakan sebuah kumpulan data yang saling berhubungan antar satu sama lain secara logika dan didesain untuk mendapatkan data yang diperlukan oleh sebuah organisasi.

## 2.16 Laravel

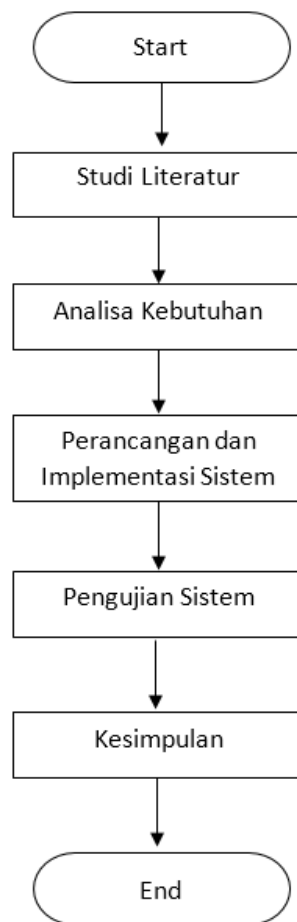
Laravel merupakan sebuah *kerangka* MVC dalam pengembangan berbasis web yang tertulis dalam bahasa PHP. Kerangka Laravel di-*design* untuk meningkatkan kualitas dari perangkat lunak dengan cara mengurangi biaya yang dikeluarkan pada pengembangan sistem dan biaya *maintenance* serta untuk meningkatkan pengalaman berkerja dengan sistem yang menyuguhkan penyampaian *syntax* dengan jelas dan akan menghemat waktu pada saat implementasi. Laravel dibuat dengan filosofi ketentuan dibandingkan dengan konfigurasi. Yang berarti laravel membuat asumsi pintar tentang apa yang Anda ingin capai, jadi laravel dapat menyelesaikan masalah yang sama namun dengan kode yang digunakan menjadi lebih sedikit.

Laravel adalah salah satu *framework* PHP yang menyajikan modularitas kode yang sebenarnya. Hal ini bisa terjadi karena adanya kombinasi antara *driver* dan bundel sistem. *Driver* dapat mengganti dan meng-*extend session*, *database*, dan fungsi *authethication*, sementara bundel dapat mengemas kode apa pun untuk digunakan kembali atau bahkan bisa dibagikan kepada pengguna laravel lainnya sehingga kita dapat memperluas jangkauan *package* yang bisa kita gunakan.

Laravel juga memberikan alat untuk berinteraksi dengan *database*. *Database migration* membuat Anda dapat dengan mudah membuat dan memodifikasi *database* dengan cara yang *independent*. *Migration* bisa dijalankan pada tipe *database* yang di-*support laravel* seperti MySQL, PostgreSQL, MSSQL, dan SQLite. Untuk cara berinteraksi dengan *database* dalam jalan *object oriented* biasa disebut Eloquent. Dengan Eloquent kita dapat membuat, mengambil, memperbarui, dan menghapus data pada *database* tanpa menuliskan kode SQL. Eloquent juga menyajikan manajemen hubungan antar objek yang kuat secara otomatis untuk Anda. Laravel juga menyajikan *command-line interface tool* yang di sebut dengan *Artisan*. Dengan *Artisan*, seorang developer bisa berinteraksi dengan sistem yang dibuat untuk melakukan sebuah aksi seperti menjalankan *migration*, menguji unit, dan menjalankan perintah yang sudah terjadwal. Laravel juga menyajikan manajemen rute URL *website* yang baik. Untuk menggantikan *inline PHP* yang lebih baik di pandang digunakan sebuah *templete view* yang disebut sebagai *Blade* (McCool, 2012).

## BAB 3 METODOLOGI

Metodologi penelitian yang digunakan oleh penulis diadaptasi dari waterfall yang dikerjakan melewati beberapa tahapan. Tahapan tersebut digambarkan pada gambar 3.1 Diagram alir metode penelitian, yang dapat dilihat di bawah ini.



Gambar 3.1 Diagram metodologi penelitian

### 3.1 Studi Literatur

Untuk melakukan penelitian ini diperlukannya sebuah tahapan yang dinamakan dengan studi literatur. Hal ini dilakukan dengan tujuan untuk memperkaya dasar teori dan mempelajari berbagai referensi yang akan berhubungan dalam pengembangan sistem. Referensi yang sudah dipelajari akan menjadi dasar ilmu pengetahuan pada tahap melakukan analisis, perancangan, dan implementasi sekaligus pengujian pada sistem. Berikut adalah beberapa bidang ilmu yang berhubungan dengan sistem pengembangan sistem penjualan dengan rekomendasi kompatibilitas komponen pada toko komputer:

1. Toko komputer :
  - a. Pendataan toko komputer.



- b. Proses bisnis jual beli komponen komputer secara *online*.
- c. *E-commerce*.

## 2. Komponen komputer :

- a. Pendataan komponen komputer.
- b. Detail spesifikasi komponen komputer.

## 3. Pengembangan perangkat lunak :

- a. Mempelajari SDLC waterfall.
- b. Pendekatan berorientasi objek.

## 4. Mempelajari teknologi dalam pengembangan sebuah perangkat lunak berupa :

- a. UML
- b. HTML
- c. PHP
- d. CSS
- e. MySQL
- f. Laravel

### 3.2 Analisis Kebutuhan

Analisis kebutuhan dilakukan untuk mengidentifikasi kebutuhan fungsional secara lengkap, yang diperlukan selama pembuatan sistem. Tahap analisis kebutuhan menggunakan pendekatan *object oriented analysis* (OOA). Tahap analisis kebutuhan terdiri dari lima tahapan, yaitu menentukan teknik elisitasi kebutuhan yang akan digunakan, mengidentifikasi aktor yang akan menggunakan sistem, mencari kebutuhan fungsional, dan membuat perancangan *use case diagram* untuk menggambarkan sistem dapat melakukan apa saja, dan membuat rancangan *use case scenario*. Pada pemodelan kebutuhan menggunakan pemodelan diagram UML dan dalam menggunakan metode analisis menggunakan pendekatan berorientasi objek.

Data yang digunakan untuk elisitasi kebutuhan didapatkan dengan menggunakan metode wawancara dengan pihak toko KiosKomputer, Bapak Faturrohmah sebagai *Marketing Manager*. Metode ini dilakukan sebanyak satu kali, dan didapat berbagai kebutuhan dari hasil wawancara tersebut (tertera pada lampiran 2). Dan penulis juga melakukan observasi terhadap sebuah *marketplace*, Tokopedia, dengan tujuan untuk mengadaptasi beberapa kebutuhan yang ada di Tokopedia. Hal ini dilakukan untuk memenuhi penggalan kebutuhan dalam pengembangan sistem penjualan dengan rekomendasi kompatibilitas komponen pada toko komputer.

### 3.3 Perancangan dan Implementasi

Pada tahaan perancangan dari sistem dilakukan sebelum aplikasi dikerjakan. Pada tahap ini, perancangan menggunakan pendekatan *Object Oriented Design* (OOD). Tahap ini dibuat rancangan secara detail bagaimana proses pengerjaan/implementasi dari aplikasi tersebut, agar dapat menjadi acuan saat dikerjakan. Terdapat empat tahapan dalam perancangan sistem, yaitu



membuat rancangan *sequence diagram* atau diagram interaksi antara aktor dan objek, membuat pemodelan klas *diagram*, membuat rancangan data berupa *entity relationship diagram* dan *physical data model*, dan membuat rancangan antarmuka dari sistem yang akan dibuat.

Implementasi sistem dikerjakan berdasarkan dari perancangan sistem yang telah selesai dibuat. Pada tahap pembuatan sistem dalam penelitian ini menggunakan pendekatan *Object Oriented Programming (OOP)*. Pada implementasi sistem menggunakan teknologi perangkat lunak seperti *framework laravel* dan *database*-nya menggunakan MySQL.

### 3.4 Pengujian Sistem

Pengujian sistem dilakukan setelah proses implementasi sistem sudah selesai. Pada tahap pengujian akan diperiksa apakah sistem yang telah dibuat dapat berjalan dengan baik, dengan tidak ada error yang muncul dan sudah sama dengan harapan. Pengujian sistem ini dibagi menjadi 3 tingkatan, yaitu :

1. Unit

Unit adalah bagian terkecil dari sebuah sistem seperti fungsi, kelas, prosedur, dan *interface*. Unit testing adalah metode yang masing-masing unit dari kode sumber diuji untuk menentukan apakah kode tersebut bekerja dengan semestinya. Metode yang digunakan untuk menguji unit adalah dengan menggunakan *Basis Path Testing*.

2. Integrasi

Pada tahap pengujian integrasi difokuskan pada pengujian gabungan unit – unit yang membentuk sebuah kesatuan fungsional. Pada tahap ini menggunakan metode basis path testing untuk menguji integrasi dari gabungan unit tersebut.

3. Validasi

Pada tahap pengujian validasi dipastikan apakah semua bagian sistem telah dikembangkan dengan benar. Validasi dikatakan berhasil ketika sistem berfungsi sesuai dengan harapan secara fungsional maupun secara nonfungsional.

### 3.5 Kesimpulan

Pada bagian ini kesimpulan harus dapat menjawab permasalahan yang telah dijelaskan pada rumusan masalah pada bab 1. Sekaligus menyertakan saran kepada penelitian selanjutnya yang mendalami pada bidang yang sama dengan penelitian ini.



## BAB 4 ANALISIS KEBUTUHAN

Tahap awal yang harus ada dalam sebuah penelitian adalah tahap analisis kebutuhan. Tahap pertama ini dibuat untuk dapat menentukan kebutuhan-kebutuhan yang harus ada dalam sistem penjualan dengan rekomendasi kompatibilitas komponen pada toko komputer. Tahap ini juga mendefinisikan aktor yang ikut serta dalam sistem dan menjelaskan interaksi yang ada, yaitu antara aktor dan sistem.

### 4.1 Elisitasi Kebutuhan Sistem

Elisitasi kebutuhan pada penelitian ini menggunakan teknik elisitasi berupa wawancara dan observasi. Narasumber untuk teknik elisitasi berupa wawancara adalah Marketing Manager KiosKomputer, yaitu Bapak Faturrohman. Proses wawancara berlangsung dengan menggunakan pertanyaan yang memiliki sifat terstruktur, yaitu pertanyaan sudah disiapkan sebelum melakukan wawancara. Berdasarkan hasil wawancara dari penulis dan pihak Marketing Manager KiosKomputer, banyak pelanggan yang datang ke toko kios komputer untuk membeli sebuah komputer namun pelanggan terlebih dahulu meminta penjelasan mengenai *part* yang telah dipilih oleh pelanggan sebelum datang ke toko komputer. Kemudian, beberapa pelanggan kios komputer didapati mengembalikan komponen komputer yang telah dibeli dengan alasan ketika digunakan di rumah komponen komputer yang telah dibeli tidak kompatibel dengan komputer yang ada di rumah. Sehingga, sistem yang akan dibuat nantinya harus bisa memberikan arahan kepada pelanggan pada saat memilih komponen komputer agar pada saat komponen komputer tersebut dirakit menjadi sebuah komputer yang utuh, komputer tersebut tidak mengalami masalah pada kompatibilitas antar komponen yang telah dibeli oleh pelanggan.

Teknik elisitasi berupa observasi dilakukan dengan mengidentifikasi salah satu *website e-commerce* yang marak digunakan oleh penduduk Indonesia, yaitu Tokopedia. Kebutuhan sistem yang diadaptasi berupa bagaimana cara pembeli bisa melihat barang yang dijual, bagaimana cara pelanggan bisa melakukan pembayaran untuk membeli sebuah produk, bagaimana cara pelanggan dapat melakukan pembayaran, bagaimana cara pihak penjual dapat menjual produknya melalui *website*, dan bagaimana cara pihak penjual mengetahui bahwa ada pembeli yang membeli sebuah produk di lapak tokonya.

### 4.2 Identifikasi Aktor

Aktor yang terlibat dalam sistem penjualan dengan rekomendasi kompatibilitas komponen pada toko komputer adalah *user*, *member*, dan toko komputer. Untuk penjelasan mengenai aktor yang ada pada sistem disajikan pada tabel 4.1.



**Tabel 4.1 Identifikasi aktor**

No.	Aktor	Deskripsi
1	<i>User</i>	<i>User</i> merupakan aktor yang memasuki sistem sebelum login atau belum terdaftar menjadi <i>member</i> dan toko komputer.
2	<i>Member</i>	<i>Member</i> merupakan <i>user</i> yang telah melakukan register dan login. <i>Member</i> dapat melihat barang, memilih barang, membeli barang, dan melakukan pembayaran terhadap barang yang telah dipilih.
3	Toko Komputer	Toko Komputer merupakan <i>user</i> yang telah melakukan <i>login</i> . Toko Komputer dapat melihat daftar pesanan, konfirmasi pembayaran, memasukkan data pengiriman barang, melihat barang jualan, menghapus barang jualan, menambah barang jualan, dan mengubah barang jualan.

### 4.3 Kebutuhan Fungsional

Dari elisitasi kebutuhan didapatkan suatu kebutuhan. Pada penelitian ini, setiap kebutuhan fungsional diberi kode fungsi berupa STMK\_XXYY. Kata STMK sendiri merupakan singkatan dari sistem toko menjual komponen. Sementara itu, XX adalah kode kebutuhan fungsional dan YY adalah kode spesifikasi dari kebutuhan fungsional tersebut. Untuk detail kebutuhan fungsional pada penelitian ini disajikan pada tabel 4.2.

**Tabel 4.2 Kebutuhan fungsional dan spesifikasi kebutuhan**

No.	Kode Fungsi	Kebutuhan Fungsional dan Spesifikasi Kebutuhan	Use Case
1	STMK_0100	Sistem harus menyediakan sebuah fungsi <i>login</i> yang dapat mengizinkan aktor untuk mengakses sistem.	<i>Login</i>
1.1	STMK_0101	Sistem harus menyediakan fungsi untuk memilih ingin <i>login</i> sebagai <i>user</i> dan Toko.	
1.2	STMK_0102	Sistem harus menyediakan <i>form</i> yang memiliki masukan data <i>e-mail</i> dan <i>password</i> .	
2	STMK_0200	Sistem harus menyediakan sebuah fungsi <i>logout</i> yang	<i>Logout</i>



		dapat mengizinkan <i>user</i> ataupun toko untuk keluar dari sistem.	
3	STMK_0300	Sistem harus menyediakan sebuah fungsi register yang dapat mengizinkan <i>user</i> mendaftar menjadi seorang <i>member</i> .	Register
3.1	STMK_0301	Sistem menyediakan formulir <i>register member</i> yang memiliki masukkan data berupa nama, <i>email</i> , <i>password</i> , dan konfirmasi <i>password</i> .	
4	STMK_0400	Sistem dapat menampilkan halaman panduan untuk pengguna mengenai penjelasan detail dari komponen-komponen komputer.	Melihat halaman panduan
5	STMK_0500	Sistem harus dapat menampilkan daftar komponen komputer.	Melihat daftar komponen
5.1	STMK_0501	Sistem harus dapat menampilkan daftar komponen berdasarkan kategori komponen yang dipilih oleh <i>member</i> .	
5.2	STMK_0502	Sistem harus dapat menampilkan nama barang, harga barang, dan deskripsi barang.	
6	STMK_0600	Sistem harus dapat menampilkan halaman simulasi pemilihan komponen komputer.	Melihat halaman simulasi
6.1	STMK_0601	Sistem harus dapat menyediakan form yang memiliki masukan data <i>processor</i> , <i>Motherboard</i> , <i>RAM</i> , <i>VGA</i> , <i>Storage</i> , <i>Power Supply</i> , dan <i>Casing</i> .	

6.2	STMK_0602	Sistem harus dapat menyediakan daftar produk sesuai kategori yang dipilih oleh pengguna sistem.	
6.3	STMK_0603	Sistem harus dapat menyimpan produk yang telah dipilih oleh <i>member</i> .	
6.4	STMK_0604	Sistem harus dapat mendeteksi apakah konsumsi daya listrik maksimal komputer dapat ditanggung oleh <i>Power Supply</i> yang telah dipilih oleh <i>member</i> .	
6.5	STMK_0605	Sistem harus dapat menampilkan total harga dari komponen komputer yang telah dipilih.	
6.6	STMK_0606	Sistem harus dapat menyediakan tombol untuk menghapus pilihan komponen yang telah dipilih dari halaman simulasi.	
7	STMK_0700	Sistem harus dapat menyediakan fitur untuk menambahkan isi dari halaman simulasi.	Mengisi halaman simulasi
7.1	STMK_0701	Sistem harus dapat menyimpan isi dari halaman simulasi.	
8	STMK_0800	Sistem harus dapat menyediakan fitur tambah barang ke keranjang.	Memasukkan barang ke Keranjang
9	STMK_0900	Sistem harus dapat menampilkan daftar barang yang ada di keranjang.	
9.1	STMK_0901	Sistem harus dapat menampilkan nama barang, harga barang, jumlah barang yang ingin dibeli, dan total harga barang yang ada di keranjang.	Melihat Keranjang



10	STMK_1000	Sistem harus dapat menyediakan fitur hapus barang yang ada di keranjang.	Menghapus isi keranjang
11	STMK_1100	Sistem harus dapat menyediakan formulir alamat pengiriman yang menerima masukan data seperti nama lengkap penerima, nomor telepon penerima, alamat, provinsi, kota, kecamatan, kelurahan, dan kode pos.	Mengisi alamat pengiriman
12	STMK_1200	Sistem harus dapat menyediakan sebuah fungsi yang dapat mengubah alamat pengiriman <i>member</i> .	Mengubah alamat pengiriman
12.1	STMK_1201	Sistem harus dapat menyediakan ubah formulir alamat pengiriman yang menerima masukan data seperti nama lengkap penerima, nomor telepon penerima, alamat, provinsi, kota, kecamatan, kelurahan, dan kode pos.	
13	STMK_1300	Sistem harus dapat menyediakan fitur pembelian barang yang telah ada di dalam keranjang.	Membeli barang
13.1	STMK_1301	Sistem harus dapat menyediakan tombol untuk membeli barang ketika ada komponen komputer di dalam keranjang.	
14	STMK_1400	Sistem harus dapat menampilkan halaman transaksi kepada <i>member</i> .	Melihat daftar transaksi
14.1	STMK_1401	Sistem harus dapat menampilkan transaksi yang sedang terjadi dan menampilkan riwayat transaksi <i>member</i> .	

14.2	STMK_1402	Sistem harus dapat menampilkan tanggal, kode transaksi, status transaksi, dan sebuah tombol "Detail".	
15	STMK_1500	Sistem harus dapat menampilkan halaman detail transaksi	
15.1	STMK_1501	Sistem harus dapat menampilkan nomor rekening toko, atas nama rekening, kode transaksi, detail pemesanan, dan detail pengiriman	Melihat detail transaksi
15.2	STMK_1502	Sistem harus dapat menampilkan status transaksi dan juga status pengiriman barang.	
16	STMK_1600	Sistem harus dapat menyediakan fitur validasi pembayaran yang dapat digunakan oleh <i>member</i> .	Validasi Pembayaran
16.1	STMK_1601	Sistem harus dapat menyediakan <i>form</i> untuk mengunggah bukti pembayaran dalam bentuk foto.	
17	STMK_1700	Sistem harus dapat menyediakan fitur untuk konfirmasi barang telah diterima.	Menerima barang
17.1	STMK_1701	Sistem akan menampilkan tombol tersebut setelah toko mengirimkan barang.	
18	STMK_1800	Sistem harus dapat menyediakan tombol yang akan ditekan oleh <i>member</i> untuk menyatakan sudah membayar pesanan.	
18.1	STMK_1801	Sistem harus dapat menampilkan bukti pembayaran yang telah	Konfirmasi Pembayaran



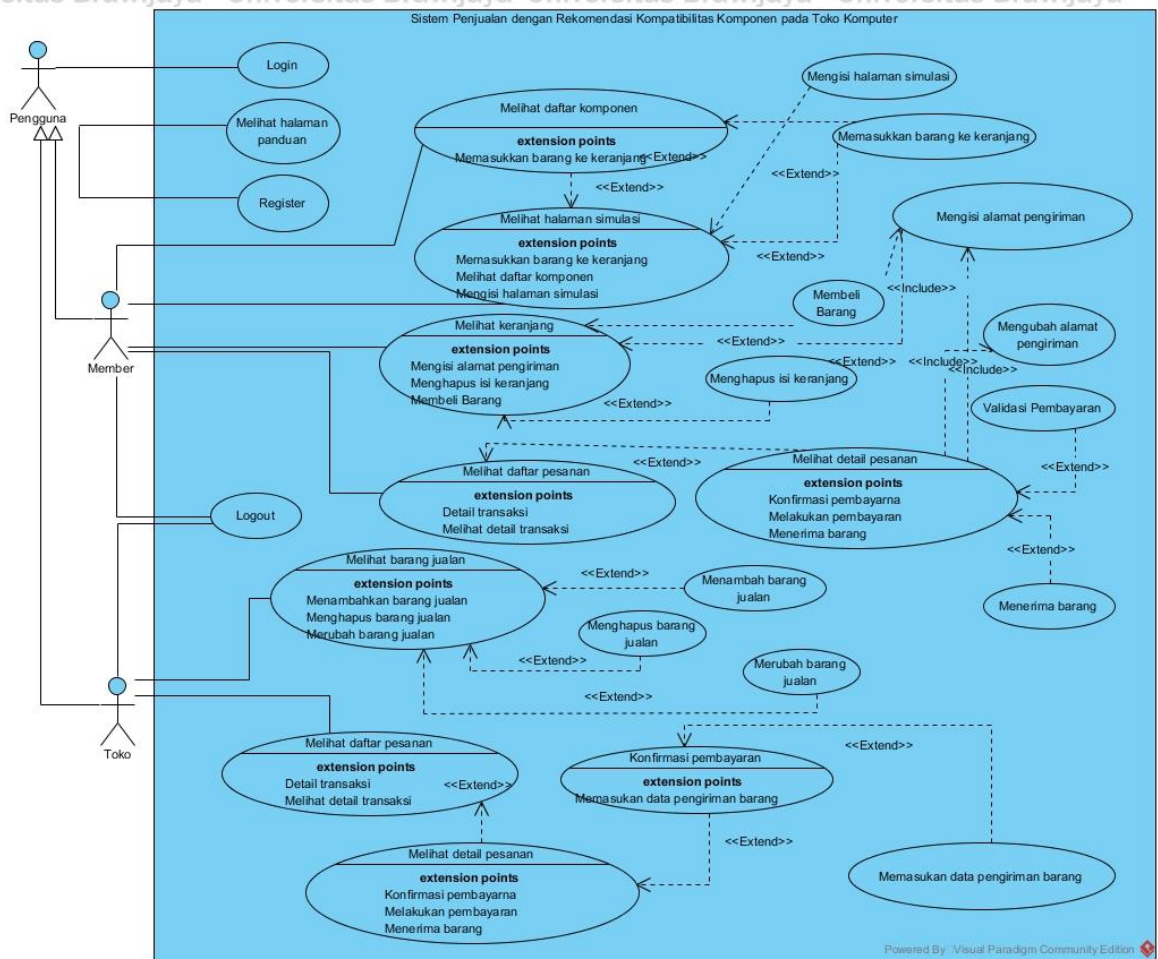
		diunggah oleh pembeli dalam bentuk foto.	
18.2	STMK_1802	Sistem harus dapat menyediakan tombol yang menyatakan bahwa pembeli telah melunasi pembayaran.	
18.3	STMK_1803	Sistem harus dapat mengubah status pembayaran yang ada di halaman detail transaksi.	
19	STMK_1900	Sistem harus dapat menampilkan daftar barang jualan.	
19.1	STMK_1901	Sistem harus dapat menampilkan daftar barang jualan berdasarkan kategori komponen komputer.	Melihat barang jualan
19.2	STMK_1902	Sistem harus dapat menampilkan tabel nama barang, detail barang, harga barang, dan stok barang.	
20	STMK_2000	Sistem harus dapat menyediakan fitur menambahkan barang jualan.	
20.1	STMK_2001	Sistem harus dapat menampilkan formulir tambah barang yang menerima masukkan data berupa nama barang, detail barang, stok barang, dan harga barang.	Menambah barang jualan
21	STMK_2100	Sistem harus dapat menyediakan fitur ubah data detail barang jualan.	Mengubah barang jualan
21.1	STMK_2101	Data yang dapat diubah adalah nama barang, harga barang, detail barang, dan stok barang.	
22	STMK_2200	Sistem harus dapat menyediakan fitur menghapus barang jualan.	Menghapus barang jualan

22.1	STMK_2201	Sistem harus dapat menghapus barang jualan yang dipilih oleh pengguna sistem.	
23	STMK_2300	Sistem harus dapat menampilkan daftar pesanan pembelian.	
23.1	STMK_2301	Sistem harus dapat menampilkan tabel nama barang, harga barang, alamat, nomor telepon, dan jumlah barang.	Melihat daftar pesanan
23.2	STMK_2302	Sistem harus dapat menampilkan status pembayaran yang telah dilakukan oleh pembeli. Status yang ditampilkan berupa "Menunggu konfirmasi pembayaran", "Menunggu pembayaran" dan "Lunas".	
24	STMK_2400	Sistem harus dapat menyediakan fitur untuk menampilkan halaman detail transaksi.	
24.1	STMK_2401	Sistem harus dapat menampilkan status pembayaran, kode transaksi, detail barang yang dibeli, dan alamat pengiriman yang telah di isikan oleh pembeli.	Melihat detail pesanan
24.2	STMK_2402	Sistem harus dapat menampilkan bukti transaksi dalam bentuk foto yang telah diunggah oleh pembeli.	
25	STMK_2500	Sistem harus dapat menyediakan fitur untuk memasukkan data pengiriman barang.	Memasukkan data pengiriman barang
25.1	STMK_2501	Sistem harus dapat menampilkan formulir kurir dan nomor resi.	



#### 4.4 Use Case Diagram

Berdasarkan kebutuhan fungsional, dibuatlah sebuah diagram *use case*. Diagram *Use case* menggambarkan aktivitas sistem yang tersajikan pada Gambar 4.1.



Gambar 4.1 Pemodelan diagram *use case* sistem penjualan komponen komputer

#### 4.5 Use case scenario

Berdasarkan use case diagram yang ada pada Gambar 4.1, dibuatlah rincian berupa use case scenario. Hal ini dilakukan untuk memberikan deskripsi berupa urutan langkah yang dijalani oleh aktor saat berinteraksi dengan sistem.

##### 1. Login

Tabel 4.3 use case scenario login

Flow of events untuk login	
Kode Fungsi	STMK_0100
Actor	User

<b>Objective</b>	User dapat masuk ke dalam sistem.
<b>Pre-Condition</b>	User masih berada di dalam halaman <i>Home</i> .
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. User menekan tombol "Login".</li> <li>2. User menekan tombol "Login sebagai <i>member</i>" atau "Login sebagai toko komputer".</li> <li>3. User mengisi kolom yang ada dengan memasukkan <i>email</i> dan <i>password</i> pada halaman <i>login</i>.</li> <li>4. User menekan tombol "Login".</li> </ol>
<b>Alternative Flow</b>	<ol style="list-style-type: none"> <li>3a. Apabila user tidak mengisi salah satu kolom pada form <i>login</i> berupa alamat <i>email</i> dan <i>password</i>, maka sistem akan menampilkan sebuah pesan peringatan yang berisi "harap isi bidang ini".</li> <li>3b. Apabila user memasukkan alamat <i>email</i> atau <i>password</i> yang tidak terdaftar pada sistem, maka sistem akan menampilkan sebuah pesan peringatan yang berisi "Email Anda belum terdaftar".</li> </ol>
<b>Post-Condition</b>	Aktor berhasil teridentifikasi sebagai <i>member</i> atau toko komputer.

## 2. Logout

**Tabel 4.4 Use case scenario logout**

<i>Flow of events untuk logout</i>	
<b>Kode Fungsi</b>	STMK_0200
<b>Aktor</b>	<i>Member</i> dan Toko Komputer
<b>Objective</b>	<i>Member</i> atau toko dapat keluar dari sistem.
<b>Pre-Condition</b>	User teridentifikasi sebagai <i>member</i> atau toko komputer.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. <i>Member</i> atau toko menekan tombol "Logout".</li> </ol>
<b>Alternative Flow</b>	-
<b>Post-Condition</b>	Aktor berhasil keluar dari sistem.

## 3. Register

**Tabel 4.5 Use case scenario register**

<i>Flow of events untuk register</i>	
<b>Kode Fungsi</b>	STMK_0300



<b>Actor</b>	<i>User</i>
<b>Objective</b>	<i>User dapat teridentifikasi sebagai member.</i>
<b>Pre-Condition</b>	<i>User berada pada halaman login.</i>
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. <i>User</i> menekan tombol “create an account” yang berada pada halaman <i>login</i>.</li> <li>2. Sistem akan menampilkan sebuah halaman <i>register</i>.</li> <li>3. <i>User</i> mengisi kolom yang ada formulir registrasi berupa nama, alamat <i>email</i>, <i>password</i>, dan konfirmasi <i>password</i>.</li> <li>4. <i>User</i> menekan tombol “Register”.</li> <li>5. Sistem akan melakukan pengecekan data dari masukan <i>user</i>.</li> </ol>
<b>Alternative Flow</b>	3a. Apabila aktor tidak mengisi salah satu kolom pada <i>form</i> register, maka sistem akan menampilkan sebuah pesan peringatan yang berisi “harap diisi form ini”.
<b>Post-Condition</b>	Sistem berhasil menyimpan data <i>user</i> .

#### 4. Melihat Halaman Panduan

**Tabel 4.6 Use case scenario halaman panduan**

<i>Flow of events</i> untuk melihat halaman panduan	
<b>Kode Fungsi</b>	STMK_0400
<b>Actor</b>	<i>User dan Member</i>
<b>Objective</b>	<i>User atau member dapat melihat halaman panduan.</i>
<b>Pre-Condition</b>	<i>User berada pada halaman utama.</i>
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. <i>User atau member</i> menekan logo “Marketplace Komputer”.</li> <li>2. Sistem akan menampilkan sebuah halaman panduan.</li> </ol>
<b>Alternative Flow</b>	-
<b>Post-Condition</b>	Sistem berhasil menampilkan halaman panduan.

#### 5. Melihat Daftar Komponen

Tabel 4.7 *Use case scenario* melihat daftar komponen

<i>Flow of events</i> untuk melihat daftar komponen	
<b>Kode Fungsi</b>	STMK_0500
<b>Actor</b>	<i>Member</i>
<b>Objective</b>	<i>Member</i> dapat melihat daftar komponen.
<b>Pre-Condition</b>	<i>Member</i> berada pada halaman utama.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. <i>Member</i> menekan tombol “Kategori Komponen”.</li> <li>2. Sistem akan menampilkan <i>list</i> kategori komponen komputer.</li> <li>3. <i>Member</i> menekan katagori yang ingin dilihat.</li> <li>4. Sistem akan menampilkan halaman daftar komponen yang sesuai dengan apa yang ditekan oleh <i>member</i> beserta data nama barang, harga barang, detail barang, dan jumlah stok.</li> </ol>
<b>Alternative Flow</b>	-
<b>Post-Condition</b>	Sistem berhasil menampilkan halaman daftar komponen.

#### 6. Melihat Halaman Simulasi

Tabel 4.8 *Use case scenario* melihat halaman simulasi

<i>Flow of events</i> untuk melihat halaman simulasi	
<b>Kode Fungsi</b>	STMK_0600
<b>Actor</b>	<i>User</i> atau <i>member</i>
<b>Objective</b>	<i>User</i> atau <i>member</i> dapat melihat halaman simulasi.
<b>Pre-Condition</b>	<i>User</i> berada pada halaman utama dan sudah teridentifikasi sebagai <i>member</i> .
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. <i>User</i> atau <i>member</i> menekan tombol “Simulasi”.</li> <li>2. Sistem akan menampilkan sebuah halaman simulasi.</li> </ol>
<b>Alternative Flow</b>	2a. Apabila <i>user</i> belum melakukan <i>login</i> sebagai <i>member</i> , maka sistem akan menampilkan halaman <i>login</i> .
<b>Post-Condition</b>	Sistem berhasil menampilkan halaman simulasi.

#### 7. Mengisi Halaman Simulasi



Tabel 4.9 Use case scenario mengisi halaman simulasi

<i>Flow of events</i> untuk mengisi halaman simulasi	
<b>Kode Fungsi</b>	STMK_0700
<b>Actor</b>	Member
<b>Objective</b>	Member dapat menambahkan isi pada halaman simulasi.
<b>Pre-Condition</b>	Member berada pada halaman simulasi.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Member menekan tombol “+” untuk menambahkan komponen pada halaman simulasi.</li> <li>2. Sistem akan menampilkan sebuah halaman daftar produk Processor.</li> <li>3. Member menekan tombol “+” pada baris daftar produk Processor yang diinginkan.</li> <li>4. Sistem akan menerima masukkan dari <i>member</i> dan menyimpan data serta menampilkan data produk pada halaman simulasi.</li> <li>5. Sistem akan menampilkan sebuah halaman daftar produk <i>Motherboard</i> yang memiliki kompatibel dengan <i>Processor</i> yang sudah dipilih.</li> <li>6. Member menekan tombol “+” pada produk yang diinginkan.</li> <li>7. Sistem akan menerima masukkan dari <i>member</i> dan menyimpan data serta menampilkan data produk pada halaman simulasi.</li> <li>8. Sistem akan menampilkan sebuah halaman daftar produk <i>RAM</i> yang memiliki kompatibel dengan <i>Processor</i> dan <i>Motherboard</i>.</li> <li>9. Member menekan tombol “+” pada produk yang diinginkan.</li> <li>10. Sistem akan menerima masukkan dari <i>member</i> dan menyimpan data serta menampilkan data produk pada halaman simulasi.</li> <li>11. Sistem akan menampilkan sebuah halaman daftar produk <i>VGA Card</i>.</li> <li>12. Member menekan tombol “+” pada produk yang diinginkan.</li> <li>13. Sistem akan menerima masukkan dari <i>member</i> dan menyimpan data serta menampilkan data produk pada halaman simulasi.</li> <li>14. Sistem akan menampilkan sebuah halaman daftar produk <i>Storage</i>.</li> </ol>

	<p>15. Member menekan tombol “+” pada produk yang diinginkan.</p> <p>16. Sistem akan menerima masukkan dari <i>member</i> dan menyimpan data serta menampilkan data produk pada halaman simulasi.</p> <p>17. Sistem akan menampilkan sebuah halaman daftar produk <i>Casing</i>.</p> <p>18. Member menekan tombol “+” pada produk yang diinginkan.</p> <p>19. Sistem akan menerima masukkan dari <i>member</i> dan menyimpan data serta menampilkan data produk pada halaman simulasi.</p> <p>20. Sistem akan menampilkan sebuah halaman daftar produk <i>Power Supply</i>.</p> <p>21. Member akan menekan tombol “+” pada produk yang diinginkan.</p> <p>22. Sistem akan menerima masukkan dari <i>member</i> dan menyimpan data serta menampilkan data pada halaman simulasi.</p>
<b>Alternative Flow</b>	-
<b>Post-Condition</b>	Sistem berhasil mengisi tabel komponen pada halaman simulasi.

#### 8. Memasukkan Barang ke Keranjang

**Tabel 4.10 Use case scenario memasukkan barang ke keranjang**

<i>Flow of events</i> untuk memasukkan barang ke keranjang	
<b>Kode Fungsi</b>	STMK_0800
<b>Actor</b>	<i>Member</i>
<b>Objective</b>	<i>Member</i> dapat memasukkan barang ke dalam keranjang.
<b>Pre-Condition</b>	<i>Member</i> berada pada halaman daftar produk komponen atau pada halaman simulasi.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. <i>Member</i> mengisikan sejumlah barang yang ingin dibeli.</li> <li>2. <i>Member</i> menekan tombol “+” pada barang yang diinginkan.</li> <li>3. Sistem akan menyimpan data serta memasukkan data produk komponen ke dalam keranjang.</li> </ol>



<b>Alternative Flow</b>	1a. Apabila <i>member</i> memasukkan jumlah barang melebihi stok barang yang tersedia, maka sistem akan menampilkan sebuah pesan peringatan yang berisi “Pembelian melebihi stok yang ada”.
<b>Post-Condition</b>	Sistem berhasil menyimpan data produk yang dipilih ke dalam keranjang.

#### 9. Melihat Keranjang

**Tabel 4.11 Use case scenario melihat ke keranjang**

<i>Flow of events</i> untuk melihat keranjang	
<b>Kode Fungsi</b>	STMK_0900
<b>Actor</b>	<i>Member</i>
<b>Objective</b>	<i>Member</i> dapat melihat halaman keranjang beserta daftar produk yang ada di dalamnya.
<b>Pre-Condition</b>	<i>Member</i> sudah masuk ke dalam sistem.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. <i>Member</i> menekan tombol “Keranjang”.</li> <li>2. Sistem akan menampilkan halaman keranjang yang berisi nama barang, detail produk barang, harga barang, jumlah barang, dan total harga.</li> </ol>
<b>Alternative Flow</b>	2a. Apabila <i>member</i> tidak memiliki barang pada halaman keranjang, maka sistem akan menampilkan pemberitahuan yang berisi “Keranjang Anda kosong”.
<b>Post-Condition</b>	Sistem berhasil menampilkan data keranjang pada halaman keranjang beserta isi keranjang.

#### 10. Menghapus Isi Keranjang

**Tabel 4.12 Use case scenario menghapus isi keranjang**

<i>Flow of events</i> untuk menghapus isi keranjang	
<b>Kode Fungsi</b>	STMK_1000
<b>Actor</b>	<i>Member</i>
<b>Objective</b>	<i>Member</i> dapat menghapus isi keranjang pada halaman keranjang.
<b>Pre-Condition</b>	<i>Member</i> berada pada halaman keranjang.

<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. <i>Member</i> menekan tombol “X” pada baris yang dipilih pada tabel produk yang berada di halaman keranjang.</li> <li>2. Sistem akan menampilkan pesan untuk mengkonfirmasi penghapusan salah satu isi keranjang.</li> <li>3. <i>Member</i> menekan tombol “OK” untuk mengkonfirmasi penghapusan salah satu isi keranjang.</li> </ol>
<b>Alternative Flow</b>	<ol style="list-style-type: none"> <li>4a. Apabila <i>member</i> menekan tombol “Cancel” saat sistem menampilkan pesan konfirmasi, maka sistem akan membatalkan penghapusan salah satu isi keranjang.</li> </ol>
<b>Post-Condition</b>	Sistem berhasil menghapus salah satu isi keranjang.

#### 11. Mengisi Alamat Pengiriman

**Tabel 4.13 Use case scenario mengisi alamat pengiriman**

<i>Flow of events</i> untuk mengisi alamat pengiriman	
<b>Kode Fungsi</b>	STMK_1100
<b>Actor</b>	<i>Member</i>
<b>Objective</b>	<i>Member</i> dapat mengisi alamat pengiriman barang.
<b>Pre-Condition</b>	<i>Member</i> berada pada halaman keranjang.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. <i>Member</i> menekan tombol “bayar”.</li> <li>2. Sistem akan menampilkan pesan konfirmasi untuk memastikan bahwa <i>member</i> benar akan membayar.</li> <li>3. <i>Member</i> menekan tombol “OK”.</li> <li>4. Sistem akan menampilkan formulir alamat pengiriman.</li> <li>5. <i>Member</i> mengisi formulir alamat pengiriman yang berisi nama lengkap penerima, nomor telepon penerima, alamat, provinsi, kota, kecamatan, kelurahan, dan kode pos.</li> <li>6. <i>Member</i> menekan tombol “Simpan Alamat Pengiriman”.</li> <li>7. Sistem akan menampilkan pesan konfirmasi untuk memastikan bahwa alamat yang diisikan sudah benar.</li> <li>8. Sistem menampilkan halaman daftar transaksi.</li> </ol>



<b>Alternative Flow</b>	<p>3a. Apabila <i>member</i> menekan tombol “Cancel”, maka sistem akan membatalkan penampilan formulir alamat pengiriman.</p> <p>5a. Apabila <i>member</i> tidak mengisikan salah satu kolom pada formulir alamat pengiriman, maka sistem akan memberikan pesan peringatan yang berisi “harap isi form bagian ini”.</p> <p>8a. Apabila <i>member</i> menekan tombol “Cancel”, maka sistem akan membatalkan penyimpanan data alamat pengiriman.</p>
<b>Post-Condition</b>	Sistem berhasil menyimpan data alamat pengiriman dan menampilkan data pada halaman detail transaksi.

## 12. Mengubah Alamat Pengiriman

**Tabel 4.14 Use case scenario mengubah alamat pengiriman**

<i>Flow of events</i> untuk mengubah alamat pengiriman	
<b>Kode Fungsi</b>	STMK_1200
<b>Actor</b>	<i>Member</i>
<b>Objective</b>	<i>Member</i> dapat mengubah alamat pengiriman barang.
<b>Pre-Condition</b>	<i>Member</i> berada pada halaman detail pesanan.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. <i>Member</i> menekan tombol “bayar”.</li> <li>2. Sistem akan menampilkan pesan konfirmasi untuk memastikan bahwa <i>member</i> benar akan membayar.</li> <li>3. <i>Member</i> menekan tombol “OK”.</li> <li>4. Sistem akan menampilkan formulir alamat pengiriman.</li> <li>5. <i>Member</i> mengisi formulir alamat pengiriman yang berisi nama lengkap penerima, nomor telepon penerima, alamat, provinsi, kota, kecamatan, kelurahan, dan kode pos.</li> <li>6. <i>Member</i> menekan tombol “Simpan Alamat Pengiriman”.</li> <li>7. Sistem akan menampilkan pesan konfirmasi untuk memastikan bahwa alamat yang diisikan sudah benar.</li> <li>8. Sistem menampilkan halaman daftar transaksi.</li> </ol>

<b>Alternative Flow</b>	<p>3a. Apabila <i>member</i> menekan tombol “Cancel”, maka sistem akan membatalkan penampilan formulir alamat pengiriman.</p> <p>5a. Apabila <i>member</i> tidak mengisikan salah satu kolom pada formulir alamat pengiriman, maka sistem akan memberikan pesan peringatan yang berisi “harap isi form bagian ini”.</p> <p>8a. Apabila <i>member</i> menekan tombol “Cancel”, maka sistem akan membatalkan penyimpanan data alamat pengiriman.</p>
<b>Post-Condition</b>	Sistem berhasil menyimpan perubahan data alamat pengiriman dan menampilkan data tersebut pada halaman detail transaksi.

### 13. Membeli Barang

**Tabel 4.15 Use case scenario membeli barang**

<i>Flow of events</i> untuk membeli barang	
<b>Kode Fungsi</b>	STMK_1300
<b>Actor</b>	<i>Member</i>
<b>Objective</b>	Pesanan memiliki status “Menunggu Pembayaran”.
<b>Pre-Condition</b>	<i>Member</i> berada pada halaman keranjang.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. <i>Member</i> menekan tombol “bayar”.</li> <li>2. Sistem akan menampilkan pesan konfirmasi untuk memastikan bahwa <i>member</i> benar akan membayar.</li> <li>3. <i>Member</i> menekan tombol “OK”.</li> <li>4. Sistem akan menampilkan formulir alamat pengiriman.</li> <li>5. <i>Member</i> mengisi formulir alamat pengiriman yang berisi nama lengkap penerima, nomor telepon penerima, alamat, provinsi, kota, kecamatan, kelurahan, dan kode pos.</li> <li>6. <i>Member</i> menekan tombol “Simpan Alamat Pengiriman”.</li> <li>7. Sistem akan menampilkan pesan konfirmasi untuk memastikan bahwa alamat yang diisikan sudah benar.</li> <li>8. Sistem menampilkan halaman daftar transaksi.</li> </ol>



<b>Alternative Flow</b>	<p>5a. Apabila <i>member</i> tidak mengisikan salah satu kolom pada formulir alamat pengiriman, maka sistem akan memberikan pesan peringatan yang berisi “harap isi form bagian ini”.</p> <p>8a. Apabila <i>member</i> menekan tombol “Cancel”, maka sistem akan membatalkan penyimpanan data alamat pengiriman.</p>
<b>Post-Condition</b>	Sistem berhasil mengubah status transaksi dari “Keranjang” menjadi “Menunggu Pembayaran”.

#### 14. Melihat Daftar Transaksi

**Tabel 4.16 Use case scenario melihat daftar transaksi**

<i>Flow of events</i> untuk melihat daftar transaksi	
<b>Kode Fungsi</b>	STMK_1400
<b>Actor</b>	<i>Member</i>
<b>Objective</b>	<i>Member</i> dapat melihat halaman daftar transaksi.
<b>Pre-Condition</b>	<i>Member</i> sudah masuk ke dalam sistem.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. <i>Member</i> menekan tombol “Transaksi”.</li> <li>2. Sistem akan menampilkan halaman daftar transaksi.</li> </ol>
<b>Alternative Flow</b>	2a. Apabila <i>member</i> belum masuk ke dalam sistem, maka sistem akan menampilkan halaman <i>login</i> .
<b>Post-Condition</b>	Sistem berhasil menampilkan halaman daftar transaksi.

#### 15. Melihat Detail Transaksi

**Tabel 4.17 Use case scenario melihat detail transaksi**

<i>Flow of events</i> untuk melihat detail pengiriman	
<b>Kode Fungsi</b>	STMK_1500
<b>Actor</b>	<i>Member</i>
<b>Objective</b>	<i>Member</i> dapat melihat halaman detail transaksi.
<b>Pre-Condition</b>	<i>Member</i> berada pada halaman daftar transaksi.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. <i>Member</i> menekan tombol “Detail”.</li> <li>2. Sistem akan menampilkan halaman detail transaksi.</li> </ol>
<b>Alternative Flow</b>	2a. Apabila <i>member</i> belum masuk ke dalam sistem, maka sistem akan menampilkan halaman <i>login</i> .

<b>Post-Condition</b>	Sistem berhasil menampilkan halaman detail transaksi.
-----------------------	---

## 16. Validasi Pembayaran

**Tabel 4.18 Use case scenario validasi pembayaran**

<i>Flow of events</i> untuk mengisi validasi pembayaran	
<b>Kode Fungsi</b>	STMK_1600
<b>Actor</b>	<i>Member</i>
<b>Objective</b>	<i>Member</i> melakukan validasi pembayaran.
<b>Pre-Condition</b>	<i>Member</i> berada pada halaman detail transaksi.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. <i>Member</i> menekan tombol “choose file” dan <i>member</i> memasukkan foto hasil bukti transfer.</li> <li>2. Sistem akan menampilkan foto yang telah diunggah oleh <i>member</i>.</li> <li>3. <i>Member</i> menekan tombol “Sudah Transfer”.</li> <li>4. Sistem akan mengubah status transaksi dari “Menunggu Pembayaran” menjadi “Menunggu Konfirmasi Pembayaran”.</li> </ol>
<b>Alternative Flow</b>	3a. Apabila <i>member</i> tidak mengisikan foto hasil bukti transfer, maka sistem akan menampilkan pesan peringatan yang berisi “Foto bukti transfer belum diupload, Mohon untuk upload terlebih dahulu”.
<b>Post-Condition</b>	Sistem berhasil menampilkan halaman detail transaksi.

## 17. Menerima Barang

**Tabel 4.19 Use case scenario menerima barang**

<i>Flow of events</i> untuk menerima barang	
<b>Kode Fungsi</b>	STMK_1700
<b>Actor</b>	<i>Member</i>
<b>Objective</b>	Sistem mengubah status pesanan dari “Sudah dikirim” menjadi “Sudah Di Terima”.
<b>Pre-Condition</b>	<i>Member</i> berada pada halaman detail transaksi dan status pengiriman adalah “Sudah dikirim”.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. <i>Member</i> menekan tombol “Sudah Menerima Barang?”.</li> </ol>



	<p>2. Sistem akan menampilkan pesan peringatan “Ketika Anda menekan tombol ini maka Transaksi Dinyatakan Selesai”.</p> <p>3. <i>Member</i> menekan tombol “OK”.</p> <p>4. Sistem menerima <i>input</i> dari aktor dan mengubah status.</p>
<b>Alternative Flow</b>	3a. Apabila <i>member</i> tombol “Cancel”, maka sistem akan membatalkan permintaan <i>member</i> .
<b>Post-Condition</b>	Sistem berhasil mengubah status pesanan dari “Sudah dikirim” menjadi “Sudah Di Terima”.

#### 18. Konfirmasi Pembayaran

**Tabel 4.20 Use case scenario konfirmasi pembayaran**

Flow of events untuk konfirmasi pembayaran	
<b>Kode Fungs</b>	STMK_1800
<b>Actor</b>	Toko Komputer
<b>Objective</b>	Sistem mengubah status pesanan dari “Menunggu Konfirmasi Pembayaran” menjadi “Lunas”.
<b>Pre-Condition</b>	Toko Komputer sudah masuk ke dalam sistem dan berada pada halaman daftar pesanan.
<b>Main Flow</b>	<p>1. Toko Komputer menekan tombol “Detail” pada pesanan yang akan dilakukan konfirmasi pembayaran.</p> <p>2. Sistem akan menampilkan halaman detail pesanan yang menampilkan foto bukti transfer beserta detail barang pesanan, dan alamat pembeli.</p> <p>1. Toko Komputer menekan tombol “Sudah Transfer”.</p> <p>2. Sistem akan mengubah status pesanan dari “Menunggu Konfirmasi Pembayaran” menjadi “Lunas”.</p>
<b>Alternative Flow</b>	3a. Apabila status pesanan adalah “Lunas”, maka sistem tidak akan menampilkan tombol “Sudah Tranfer”.
<b>Post-Condition</b>	Sistem berhasil mengubah status pesanan dari “Menunggu Konfirmasi Pembayaran” menjadi “Lunas”.

#### 19. Melihat Barang Jualan

Tabel 4.21 *Use case scenario* melihat barang jualan

<i>Flow of events</i> untuk melihat barang jualan	
<b>Kode Fungsi</b>	STMK_1900
<b>Actor</b>	Toko Komputer
<b>Objective</b>	Toko Komputer dapat melihat halaman barang jualan.
<b>Pre-Condition</b>	Toko Komputer sudah masuk ke dalam sistem.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Toko Komputer menekan tombol “Your Product”</li> <li>2. Toko Komputer menekan tombol pada salah satu kategori yang ingin dilihat.</li> <li>3. Sistem akan menampilkan halaman barang jualan berdasarkan kategori yang dipilih oleh Toko Komputer.</li> </ol>
<b>Alternative Flow</b>	3a. Apabila Toko Komputer belum masuk ke dalam sistem, maka sistem akan menampilkan halaman <i>login</i> toko..
<b>Post-Condition</b>	Sistem berhasil menampilkan data barang jualan.

## 20. Menambah Barang Jualan

Tabel 4.22 *Use case scenario* menambah barang jualan

<i>Flow of events</i> untuk menambah barang jualan	
<b>Kode Fungsi</b>	STMK_2000
<b>Actor</b>	Toko Komputer
<b>Objective</b>	Toko Komputer dapat menambah barang jualan.
<b>Pre-Condition</b>	Toko Komputer berada di halaman tambah produk komponen.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Toko Komputer mengisi formulir tambah barang jualan yang berisi nama barang, detail barang, harga barang, dan stok barang.</li> <li>2. Toko Komputer menekan tombol “Tambah Barang”.</li> <li>3. Sistem akan menerima <i>input</i> data barang jualan dari Toko Komputer.</li> </ol>
<b>Alternative Flow</b>	1a. Apabila Toko Komputer tidak mengisi salah satu kolom yang ada pada <i>form</i> tambah barang jualan, maka sistem akan menampilkan pesan peringatan yang berisi “harap isi bidang ini”.



<b>Post-Condition</b>	Sistem berhasil menyimpan data barang jualan.
-----------------------	---

## 21. Mengubah Barang Jualan

**Tabel 4.23 Use case scenario mengubah barang jualan**

<i>Flow of events</i> untuk mengubah barang jualan	
<b>Kode Fungsi</b>	STMK_2100
<b>Actor</b>	Toko Komputer
<b>Objective</b>	Toko Komputer dapat mengubah barang jualan.
<b>Pre-Condition</b>	Toko Komputer berada di halaman daftar barang jualan.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Toko Komputer menekan tombol “edit” pada salah satu data barang jualan.</li> <li>2. Sistem akan menampilkan formulir untuk mengubah data barang jualan yang berisi nama barang, detail barang, stok barang, dan harga barang.</li> <li>3. Toko Komputer memasukkan data terbaru pada <i>form</i> perubahan data barang jualan.</li> <li>4. Toko Komputer menekan tombol “Update data”</li> <li>5. Sistem akan menerima <i>input</i> perubahan data barang jualan dari Toko Komputer.</li> </ol>
<b>Alternative Flow</b>	3a. Apabila Toko Komputer tidak mengisikan salah satu kolom yang ada pada <i>form</i> perubahan barang jualan, maka sistem akan menampilkan pesan peringatan yang berisi “harap isi bidang ini”.
<b>Post-Condition</b>	Sistem berhasil menyimpan perubahan data barang jualan.

## 22. Menghapus Barang Jualan

**Tabel 4.24 Use case scenario menghapus barang jualan**

<i>Flow of events</i> untuk menghapus barang jualan	
<b>Kode Fungsi</b>	STMK_2200
<b>Actor</b>	Toko Komputer
<b>Objective</b>	Toko Komputer dapat menghapus barang jualan.
<b>Pre-Condition</b>	Toko Komputer berada di halaman daftar barang jualan.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Toko Komputer menekan tombol “Hapus” pada salah satu data barang jualan.</li> </ol>

	2. Sistem akan menampilkan sebuah pesan untuk mengkonfirmasi penghapusan barang jualan. 3. Toko Komputer menekan tombol “Yes” untuk mengkonfirmasi penghapusan barang jualan. 4. Sistem akan menerima <i>input</i> dari Toko Komputer.
<b>Alternative Flow</b>	3a. Apabila Toko Komputer menekan tombol “No”, maka sistem akan membatalkan penghapusan barang jualan.
<b>Post-Condition</b>	Sistem berhasil menghapus data barang jualan.

#### 23. Melihat Daftar Pesanan

**Tabel 4.25 Use case scenario melihat daftar pesanan**

<i>Flow of events</i> untuk melihat daftar pesanan	
<b>Kode Fungsi</b>	STMK_2300
<b>Actor</b>	Toko Komputer
<b>Objective</b>	Toko Komputer dapat melihat halaman daftar pesanan.
<b>Pre-Condition</b>	Toko Komputer sudah masuk ke dalam sistem.
<b>Main Flow</b>	1. Toko Komputer menekan tombol “Pesanan”. 2. Sistem akan menampilkan halaman daftar pesanan yang berisi tanggal pesanan, kode transaksi, status pesanan, dan jumlah harga.
<b>Alternative Flow</b>	2a. Apabila Toko Komputer belum masuk ke dalam sistem, maka sistem akan menampilkan halaman <i>login</i> .
<b>Post-Condition</b>	Sistem berhasil menampilkan halaman daftar pesanan.

#### 24. Melihat Detail Pesanan

**Tabel 4.26 Use case scenario melihat detail pesanan**

<i>Flow of events</i> untuk melihat detail pesanan	
<b>Kode Fungsi</b>	STMK_2400
<b>Actor</b>	Toko Komputer
<b>Objective</b>	Toko Komputer dapat melihat halaman detail pesanan.
<b>Pre-Condition</b>	Toko Komputer berada di halaman daftar pesanan.



<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Toko Komputer menekan tombol “Detail”.</li> <li>2. Sistem akan menampilkan halaman detail pesanan yang berisi kode transaksi, status pesanan, status pengiriman, dan detail barang yang dipesan beserta alamat pembeli.</li> </ol>
<b>Alternative Flow</b>	1a. Apabila tidak ada pesanan yang masuk, maka sistem tidak akan menampilkan tombol “Detail”.
<b>Post-Condition</b>	Sistem berhasil menampilkan halaman detail pesanan.

#### 25. Memasukkan Data Pengiriman Barang

**Tabel 4.27 Use case scenario memasukkan data pengiriman barang**

<i>Flow of events</i> untuk memasukkan data pengiriman barang	
<b>Kode Fungsi</b>	STMK_2500
<b>Actor</b>	Toko Komputer
<b>Objective</b>	Toko Komputer dapat mengisi data pengiriman barang.
<b>Pre-Condition</b>	Toko Komputer berada pada halaman detail pesanan.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Toko Komputer menekan tombol “Sudah kirim barang?”.</li> <li>2. Sistem akan menampilkan formulir data pengiriman barang yang berisi nama kurir dan nomor resi.</li> <li>3. Toko Komputer mengisi <i>form</i> data pengiriman barang.</li> <li>4. Sistem akan menerima <i>input</i> dari Toko Komputer.</li> </ol>
<b>Alternative Flow</b>	3a. Apabila Toko Komputer tidak mengisi salah satu kolom pada <i>form</i> data pengiriman barang, maka sistem akan menampilkan pesan peringatan yang berisi “mohon isi bidang ini”.
<b>Post-Condition</b>	Sistem berhasil menyimpan data pengiriman barang.

## BAB 5 PERANCANGAN DAN IMPLEMENTASI

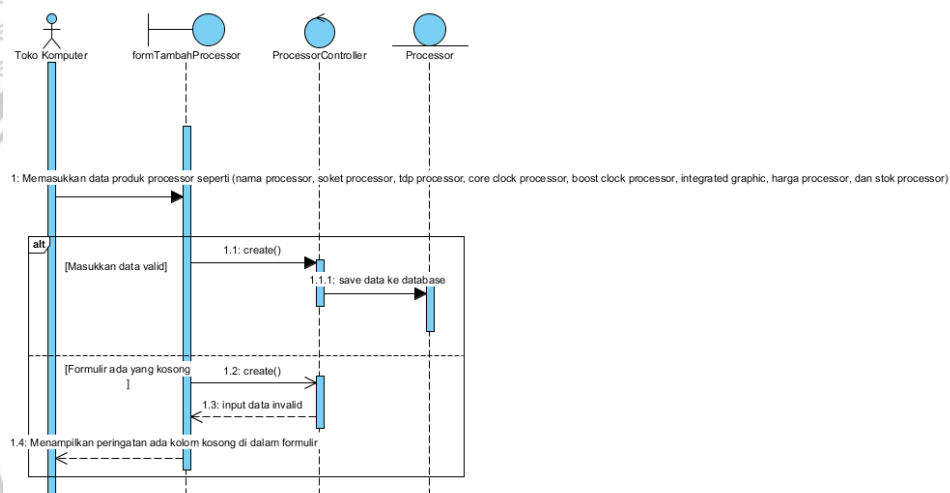
### 5.1 Perancangan Sistem

Pada tahap perancangan sistem, terdiri dari 5 tahapan berupa perancangan diagram *sequence*, perancangan klas diagram, perancangan data, komponen, dan perancangan antarmuka.

#### 5.1.1 Perancangan *Sequence* diagram

Pada tahap perancangan *sequence* diagram, dijelaskan 3 diagram *sequence*, yaitu *sequence* diagram pembelian barang, mengisi halaman simulasi, dan validasi pembayaran.

##### 5.1.1.1 *Sequence* Diagram Menambah Barang Jualan (*Processor*)



Gambar 5.1 *Sequence* manambah barang jualan (*Processor*)

Dalam sebuah *sequence* menambah barang jualan seperti Gambar 5.1, toko komputer mengeklik tombol tambah *processor* pada halaman daftar produk, kemudian *boundary* produkProcessor menggunakan fungsi tambahProcessor kepada *controller* ProcessorController untuk menggunakan *method* viewTambahProcessor() agar sistem dapat menampilkan halaman formTambahProcessor. Kemudian, Toko Komputer memasukkan data nama *processor*, soket *processor*, tdp *processor*, core clock *processor*, boost clock *processor*, integrated graphic, harga *processor* dan stok *processor*. Setelah itu, *boudary form* TambahProcessor memanggil *method* create() pada *controller* ProcessorController untuk memvalidasi masukan data dari *form*.

#### 1. *Alternative* pertama (Ketika memasukkan data valid)

Bila masukan data toko komputer dinyatakan valid maka *method* create() akan menyimpan data masukan tersebut ke dalam *entity processor*. Kemudian, *entity processor* mengembalikan hasil *query* kepada *controller*



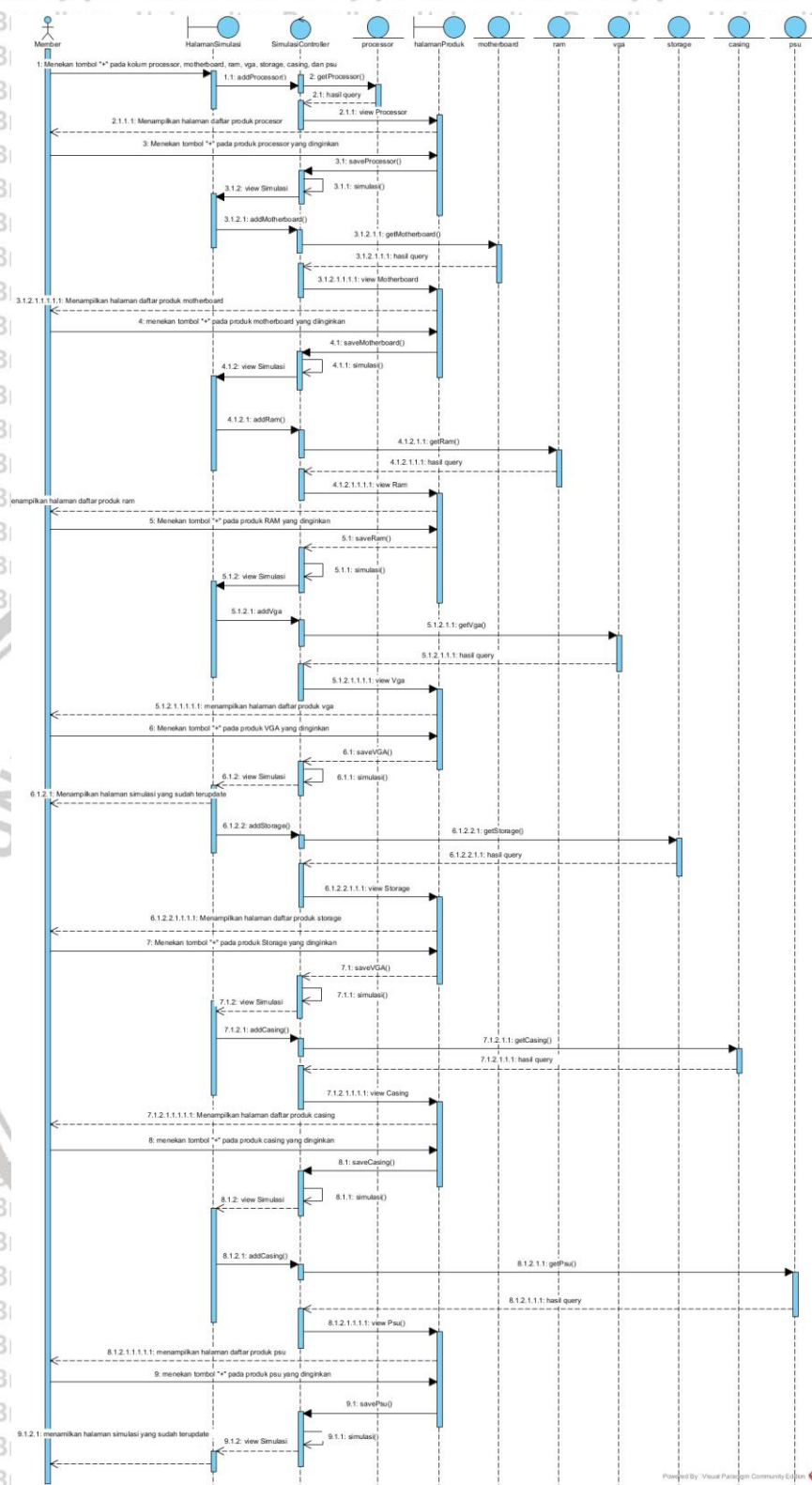
ProcessorController, sekaligus menampilkan hasil daftar produk *processor* yang terbaru.

## 2. *Alternative* kedua (Ketika data formulir ada yang kosong)

Bila masukan data toko komputer dinyatakan tidak *valid* atau ada yang kosong, maka *Controller Processor* mengembalikan status data *invalid* kepada *boundary form* TambahProcessor dan sistem memberikan respons berupa pesan peringatan pada bagian kolom formulir yang tidak terisi.

### 5.1.1.2 *Sequence Diagram* Mengisi Halaman Simulasi





**Gambar 5.2 Sequence mengisi halaman simulasi**

Dalam sebuah *sequence* mengisi halaman simulasi pada Gambar 5.2, *Member* menekan tombol "+" pada baris *processor*. Setelah itu, *boundary* HalamanSimulasi memanggil *method* *addProcessor()* pada *controller* Simulasi



Controller untuk memanggil *boundary* halamanProduk dengan data dari hasil *query* dari *entity processor*. Selanjutnya, sistem akan menampilkan halaman daftar produk *processor*. Kemudian, *member* dapat menekan tombol “+” pada baris produk *processor* yang ingin dipilih. Lalu, *boundary* halamanProduk akan memanggil *method* *saveProcessor()* yang berada pada *controller* *SimulasiController*, dan kemudian *controller* *SimulasiController* memanggil *method* bernama *simulasi()* yang akan menampilkan *viewSimulasi*. Setelah itu sistem akan menampilkan halaman simulasi yang sudah diperbaharui dengan data *processor* yang telah dipilih oleh *member*.

*Member* menekan tombol “+” pada baris *motherboard*. Setelah itu, *boundary* HalamanSimulasi memanggil *method* *addMotherboard()* pada *controller* *Simulasi Controller* untuk memanggil *boundary* halamanProduk dengan data dari hasil *query* dari *entity motherboard*. Selanjutnya, sistem akan menampilkan halaman daftar produk *motherboard*. Kemudian, *member* dapat menekan tombol “+” pada baris produk *motherboard* yang ingin dipilih. Lalu, *boundary* halamanProduk akan memanggil *method* *saveMotherboard()* yang berada pada *controller* *SimulasiController*, dan kemudian *controller* *SimulasiController* memanggil *method* bernama *simulasi()* yang akan menampilkan *viewSimulasi*. Setelah itu sistem akan menampilkan halaman simulasi yang sudah diperbaharui dengan data *processor* dan *motherboard* yang telah dipilih oleh *member*.

*Member* menekan tombol “+” pada baris ram. Setelah itu, *boundary* HalamanSimulasi memanggil *method* *addRam()* pada *controller* *Simulasi Controller* untuk memanggil *boundary* halamanProduk dengan data dari hasil *query* dari *entity* RAM. Selanjutnya, sistem akan menampilkan halaman daftar produk ram. Kemudian, *member* dapat menekan tombol “+” pada baris produk ram yang ingin dipilih. Lalu, *boundary* halamanProduk akan memanggil *method* *saveRam()* yang berada pada *controller* *SimulasiController*, dan kemudian *controller* *SimulasiController* memanggil *method* yang bernama *simulasi()* yang akan menampilkan *viewSimulasi*. Setelah itu sistem akan menampilkan halaman simulasi yang sudah diperbaharui dengan data *processor*, *motherboard*, dan RAM yang telah dipilih oleh *member*.

*Member* menekan tombol “+” pada baris VGA card. Setelah itu, *boundary* HalamanSimulasi memanggil *method* *addVga()* pada *controller* *Simulasi Controller* untuk memanggil *boundary* halamanProduk dengan data dari hasil *query* dari *entity vga*. Selanjutnya, sistem akan menampilkan halaman daftar produk VGA card. Kemudian, *member* dapat menekan tombol “+” pada baris produk VGA yang ingin dipilih. Lalu, *boundary* halamanProduk akan memanggil *method* *saveVga()* yang berada pada *controller* *SimulasiController*, dan kemudian *controller* *SimulasiController* memanggil *method* yang berada di dalam satu *class* yang sama yang bernama *simulasi()* yang akan menampilkan *viewSimulasi*. Setelah itu sistem akan menampilkan halaman simulasi yang sudah diperbaharui dengan data *processor*, *motherboard*, RAM, dan VGA yang telah dipilih oleh *member*.



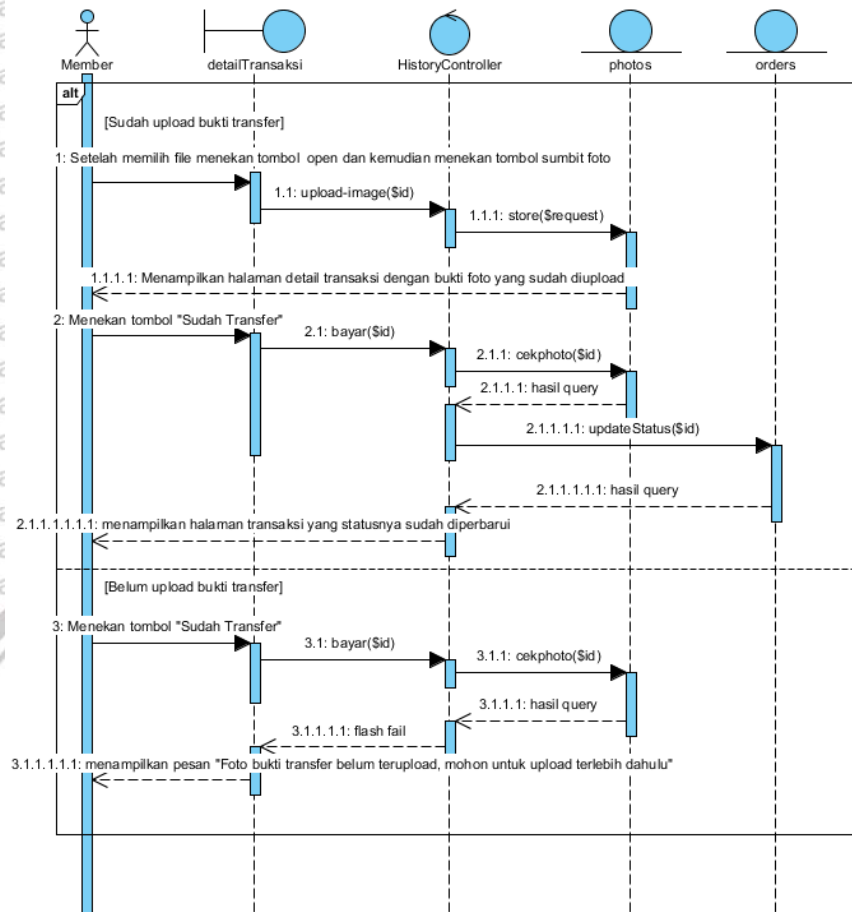
*Member* menekan tombol “+” pada baris *storage*. Setelah itu, *boundary* HalamanSimulasi memanggil *method addStorage()* pada *controller* Simulasi Controller untuk memanggil *boundary* halamanProduk dengan data dari hasil *query* dari *entity storage*. Selanjutnya, sistem akan menampilkan halaman daftar produk *storage*. Kemudian, *member* dapat menekan tombol “+” pada baris produk *storage* yang ingin dipilih. Lalu, *boundary* halamanProduk akan memanggil *method saveStorage()* yang berada pada *controller* SimulasiController, dan kemudian *controller* SimulasiController memanggil *method* yang bernama *simulasi()* kemudian menampilkan *viewSimulasi*. Setelah itu sistem akan menampilkan halaman simulasi yang sudah diperbaharui dengan data *processor*, *motherboard*, RAM, VGA, dan *storage* yang telah dipilih oleh *member*.

*Member* menekan tombol “+” pada baris *Casing*. Setelah itu, *boundary* HalamanSimulasi memanggil *method addCasing()* pada *controller* Simulasi Controller untuk memanggil *boundary* halamanProduk dengan data dari hasil *query* dari *entity casing*. Selanjutnya, sistem akan menampilkan halaman daftar produk *casing*. Kemudian, *member* dapat menekan tombol “+” pada baris produk *casing* yang ingin dipilih. Lalu, *boundary* halamanProduk akan memanggil *method saveCasing()* yang berada pada *controller* SimulasiController, dan kemudian *controller* SimulasiController memanggil *method* yang bernama *simulasi()* yang akan menampilkan *viewSimulasi*. Setelah itu sistem akan menampilkan halaman simulasi yang sudah diperbaharui dengan data *processor*, *motherboard*, RAM, VGA, *storage*, dan *casing* yang telah dipilih oleh *member*.

*Member* menekan tombol “+” pada baris *Power Supply*. Setelah itu, *boundary* HalamanSimulasi memanggil *method addPsu()* pada *controller* Simulasi Controller untuk memanggil *boundary* halamanProduk dengan data dari hasil *query* dari *entity psu*. Selanjutnya, sistem akan menampilkan halaman daftar produk *psu*. Kemudian, *member* dapat menekan tombol “+” pada baris produk *psu* yang ingin dipilih. Lalu, *boundary* halamanProduk akan memanggil *method savePsu()* yang berada pada *controller* SimulasiController, dan kemudian *controller* SimulasiController memanggil *simulasi()* yang akan menampilkan *viewSimulasi*. Setelah itu sistem akan menampilkan halaman simulasi yang sudah diperbaharui dengan data *processor*, *motherboard*, RAM, VGA, *storage*, *casing*, dan *psu* yang telah dipilih oleh *member*.



### 5.1.1.3 Sequence Diagram Validasi Pembayaran



**Gambar 5.3 Sequence validasi pembayaran**

Dalam sebuah *sequence* validasi pembayaran pada Gambar 5.3. *member* menekan tombol “choose file” yang berada pada *boundary* detailTransaksi. Kemudian, *boundary* detailTransaksi menampilkan *form* dalam bentuk pemilihan *file* foto yang ingin diunggah. Setelah memilih foto, *Member* menekan tombol *submit* foto pada *boundary* detailTransaksi yang akan memanggil *method* *upload-image(\$id)* pada *controller* *HistoryController*. Setelah itu, *controller* *HistoryController* menggunakan *method* *store(\$request)* kepada *entity* *photos* untuk menyimpan foto yang diunggah oleh *member* ke dalam *database*. Lalu, sistem menampilkan halaman detailTransaksi sekaligus menampilkan bukti transfer yang telah diunggah oleh *member*. Kemudian, aktor menekan tombol “Sudah Transfer” pada *boundary* detailTransaksi yang akan memanggil *method* *bayar(\$id)* pada *controller* *HistoryController* untuk menggunakan *method* *cekphoto(\$id)* kepada *entity* *photos*. Setelah itu, *entity* *photos* mengembalikan *hasil query* kepada *controller* *HistoryController*. Bila foto terdeteksi sudah diunggah maka *controller* *HistoryController* akan memanggil *method* *updateStatus(\$id)* kepada *entity* *orders*. Setelah itu, *entity* *orders* akan mengembalikan *hasil query* kepada *HistoryController* yang kemudian

menampilkan halaman transaksi yang statusnya sudah diperbarui menjadi “Menunggu Konfirmasi Pembayaran”.

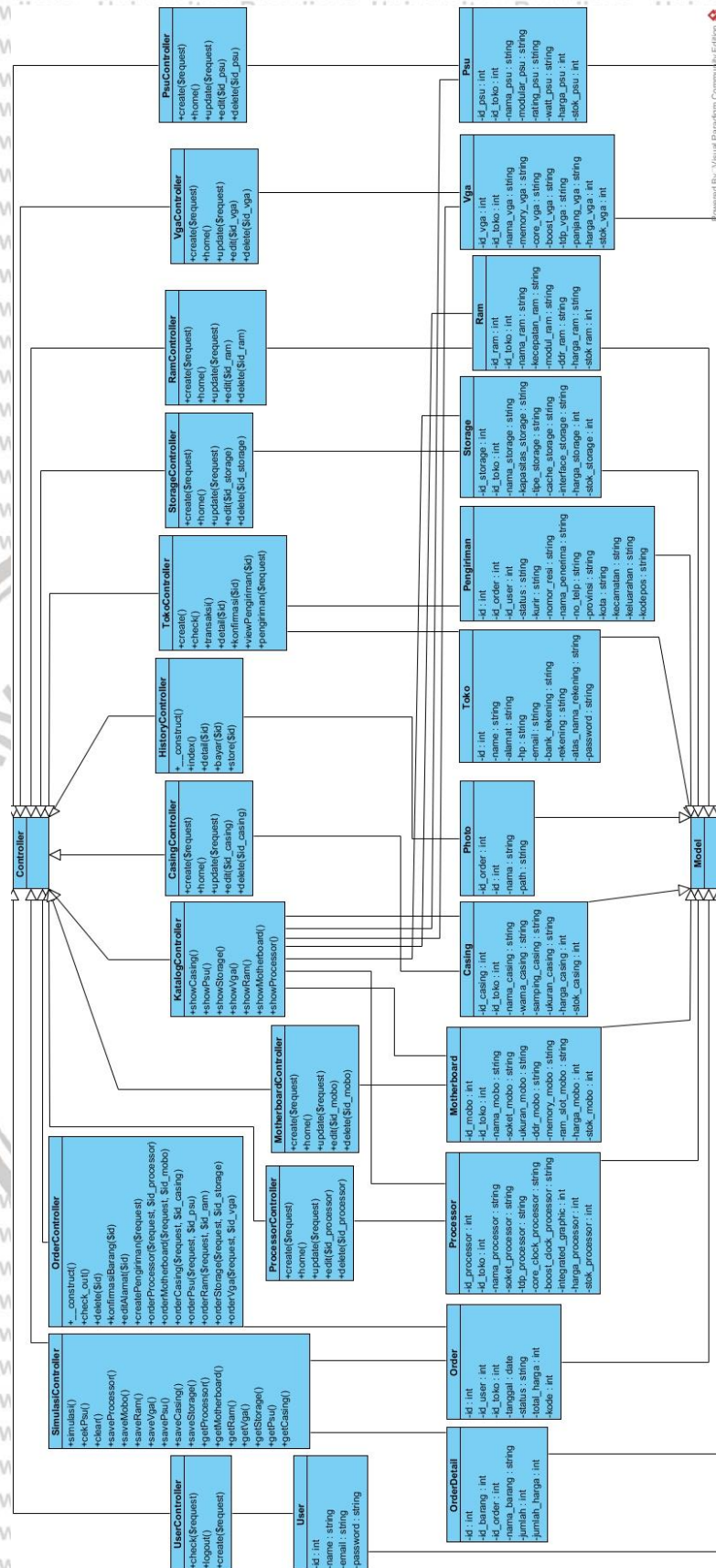
Pada sebuah *sequence* validasi pembayaran pada jalur alternatif bila member belum *upload* bukti transfer, member menekan tombol “Sudah Transfer” yang memanggil *method* bayar(\$id) kepada *controller* HistoryController. Setelah itu, HistoryController menggunakan *method* cekphoto(\$id) pada *entity* photos yang sekaligus mengembalikan hasil *query* kepada *controller* HistoryController. Controller HistoryController akan memanggil *method* flash fail terhadap *boundary* detailTransaksi yang akan menampilkan pesan “Belum upload bukti transfer”.

### 5.1.2 Perancangan Class Diagram

Perancangan diagram kelas dibuat dengan tujuan untuk memberikan gambaran mengenai hubungan antara tiap *class* yang ada didalam pengembangan sistem dengan rekomendasi kompatibilitas komponen pada toko komputer. Adapun perancangan class diagram tersajikan pada Gambar 5.4.







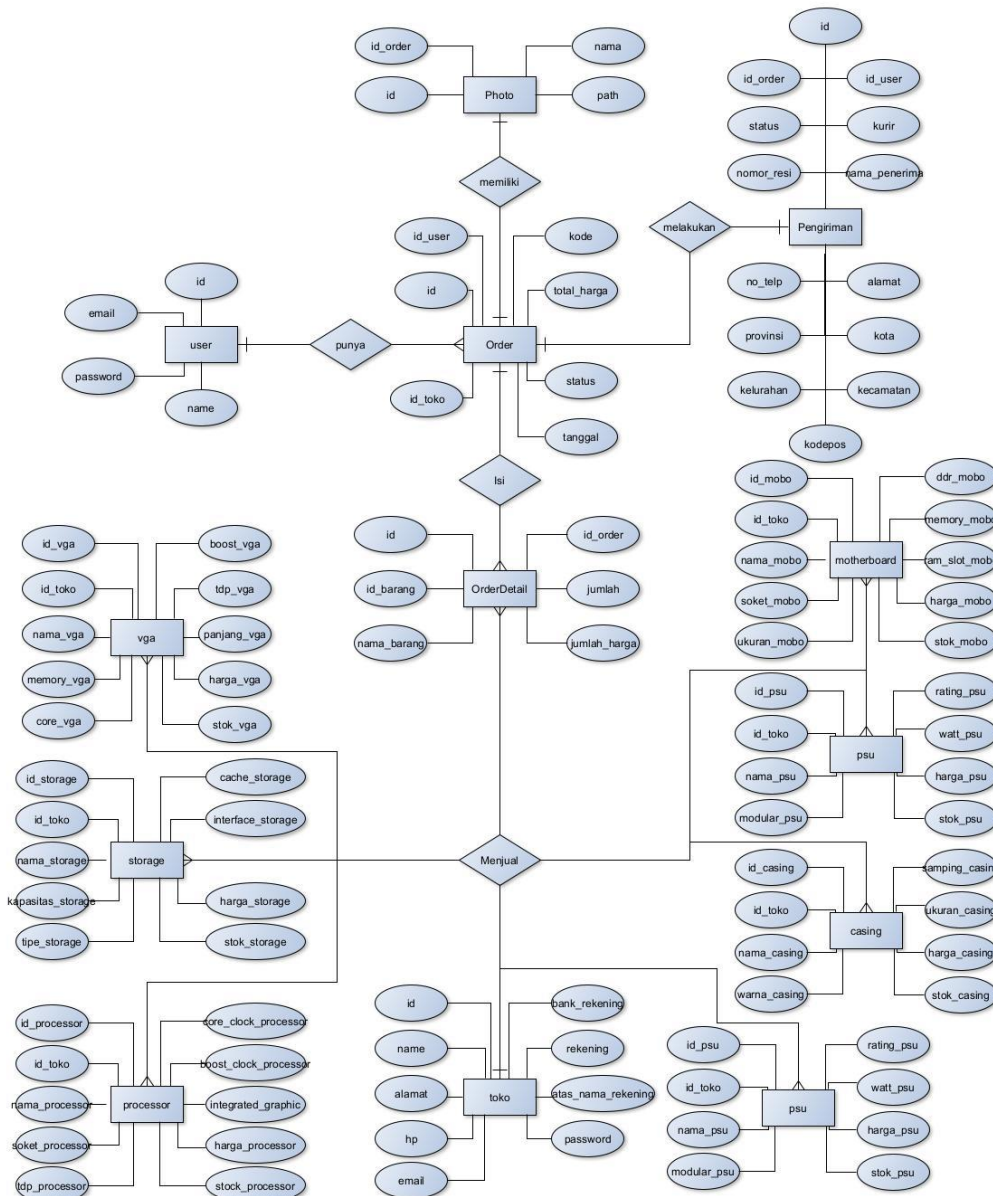
Gambar 5.4 Rancangan umum *class* diagram

### 5.1.3 Perancangan Data

Pada tahapan perancangan data dalam penelitian ini dilakukan dengan memodelkan perancangan *Entity Relationship Diagram* atau biasa disingkat menjadi ERD dan perancangan *Physical data model* yang bertujuan untuk menggambarkan *entity* memiliki data apa saja.

#### 5.1.3.1 Perancangan ERD

Berikut merupakan diagram *Entity Relationship Diagram* dari Sistem penjualan dengan Rekomendasi Kompatibilitas Komponen pada Toko Komputer yang disajikan pada gambar 5.5.

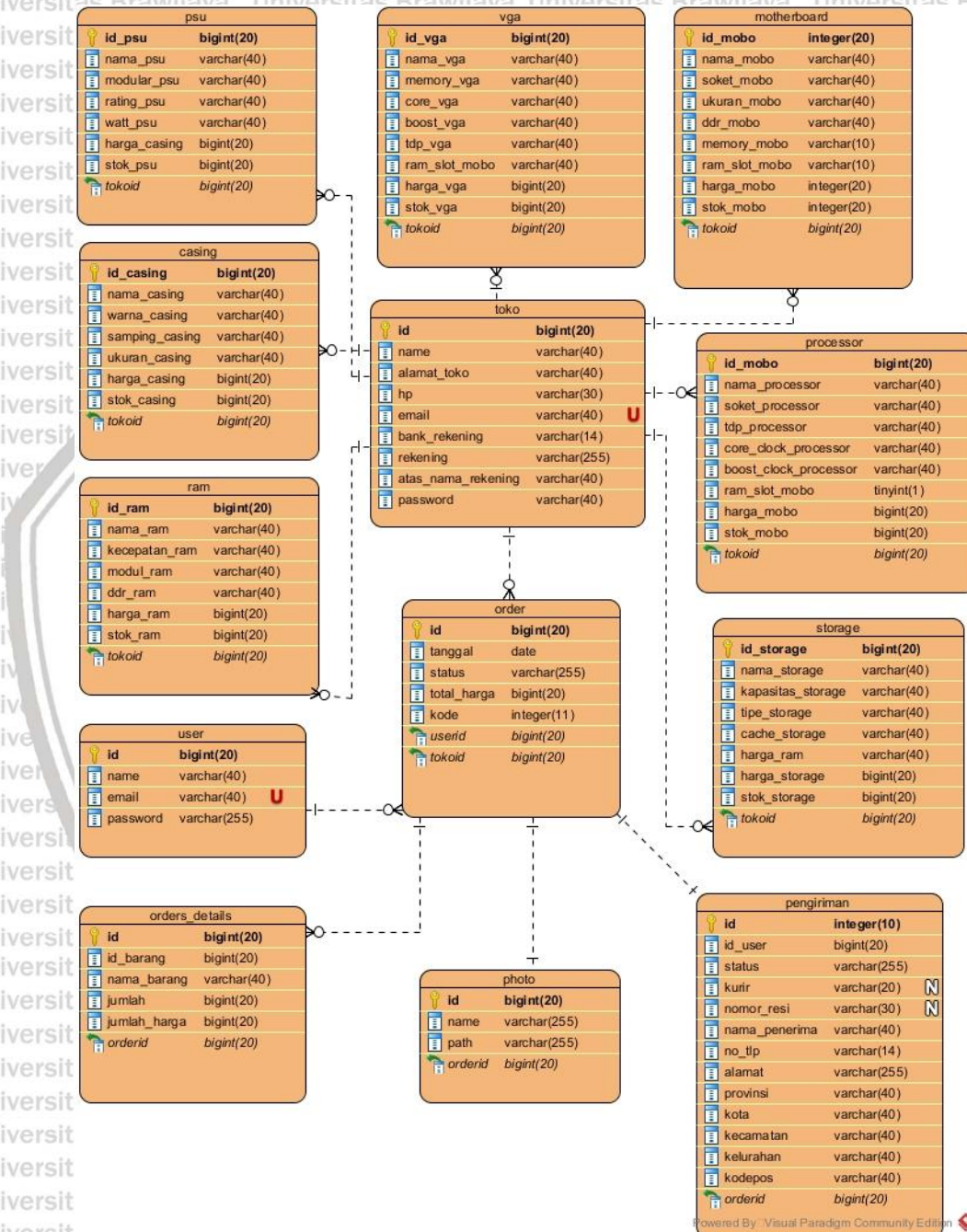




Gambar 5.5 Entity Relationship Diagram

### 5.1.3.2 Perancangan Data Physical Data Model

Pada tahap perancangan data, digunakan metode permodelan *physical data* yang hasilnya memiliki 13 tabel model yang tersajikan pada Gambar 5.6.



Gambar 5.6 Rancangan physical data model

Berdasarkan tabel *order*, terdapat 4 tabel yang memiliki relasi terhadap tabel *order*, yaitu tabel *user*, *orders\_details*, *pengiriman*, dan *photo*. Pada bagian *model*, *user* berhubungan relasi *one to many* kepada *order* sehingga satu *user*



dapat memiliki banyak *order*. Namun, pada tabel pengiriman memiliki relasi *one to one* terhadap *order* sehingga satu *order* hanya boleh memiliki satu pengiriman saja. Sedangkan pada bagian tabel toko didapati 8 tabel memiliki hubungan *one to many*. Sehingga satu toko dapat memiliki banyak *processor, motherboard, ram, vga, psu, order, casing, storage*.

#### 5.1.4 Perancangan Komponen

Pada tahapan perancangan komponen, digambarkan sebuah bagian dari sistem setiap komponen yang berada pada sistem. Hal ini dilakukan dengan melakukan perancangan komponen harus menjelaskan struktur data pada semua objek lokal dan bagian algoritma pada proses yang terjadi di dalam tingkatan komponen. Pada tahapan ini diambil tiga *class* yaitu *class processorController*, *SimulasiController*, dan *HistoryController*.

##### 5.1.4.1 Perancangan Komponen *Class ProcessorController*

Pada *class model Processor* memanggil method *create(\$request)* untuk menambahkan sebuah *processor* ke dalam *database*. Berikut merupakan hasil dari perancangan komponen *method create(\$request)* yang dapat dilihat di tabel 5.1.

Tabel 5.1 Perancangan komponen *create(\$request)*

No	Pseudocode
1	Start
2	Input data processor
3	Validate input
4	If (validate) {
5	create new processor
6	data processor = input data
7	} else {
8	Return Error Message invalid input
9	}
10	End if
11	End

##### 5.1.4.2 Perancangan Komponen *Class SimulasiController*

*Class SimulasiController* dirancang menggunakan method *simulasi()*. Untuk melakukan perhitungan total harga dan total konsumsi daya listrik dari komponen yang sudah dipilih pada halaman simulasi. Untuk penjelasan mengenai method *simulasi()* dapat dilihat di tabel 5.2.

Tabel 5.2 Perancangan komponen *simulasi()*

No	Pseudocode



1	Start
2	\$session data komponen
3	\$total harga = session data (harga_processor +
4	harga_motherboard + harga_ram + harga_vga +
5	harga_storage + harga_psu + harga_casing)
6	
7	\$tdp_komputer = session data (tdp_processor +
8	tdp_vga) + 150
9	
10	Return View Simulasi
11	End

#### 5.1.4.3 Perancangan Komponen Class HistoryController

Pada class HistoryController perancangan komponen menggunakan *method* bayar(\$id). Untuk penjelasan mengenai method bayar(\$id) dapat dilihat di tabel 5.3.

**Tabel 5.3 Perancangan komponen bayar(\$id)**

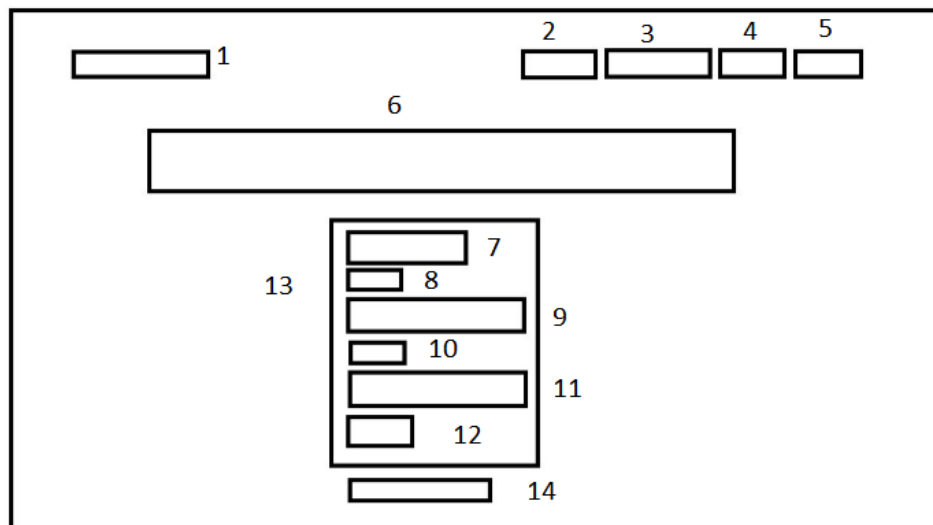
No	Pseudocode
1	Start
2	\$id_order
3	\$photo = photo where (ide_photo = ide_order)
4	if (\$photo is empty){
5	echo "Foto Bukti Transfer Belum diUpload,
6	Mohon untuk Upload bukti transfer terlebih
7	dahulu"
8	} Else
9	update status order = "Menunggu Konfirmasi
10	Pembayaran"
11	}
12	endif
13	end

#### 5.1.5 Perancangan Antarmuka

Berikut adalah bagian dari hasil perancangan antarmuka yang berada pada Sistem Penjualan dengan Rekomendasi Kompatibilitas Komponen pada Toko Komputer. Didapati menghasilkan 14 perancangan antarmuka dalam sistem ini.

#### 5.1.5.1 Perancangan Antarmuka Login User

Dibawah ini adalah rancangan dari halaman *login* untuk *user* sehingga *user* dapat masuk ke dalam sistem dengan menjadi *member* yang tersajikan pada gambar 5.7.



**Gambar 5.7 Rancangan halaman antarmuka login user**

Pada tahap perancangan antarmuka halaman login user didapati menghasilkan 14 objek yang keterangan dan tipe dari objek disajikan pada tabel 5.4.

**Tabel 5.4 Objek antarmuka login user**

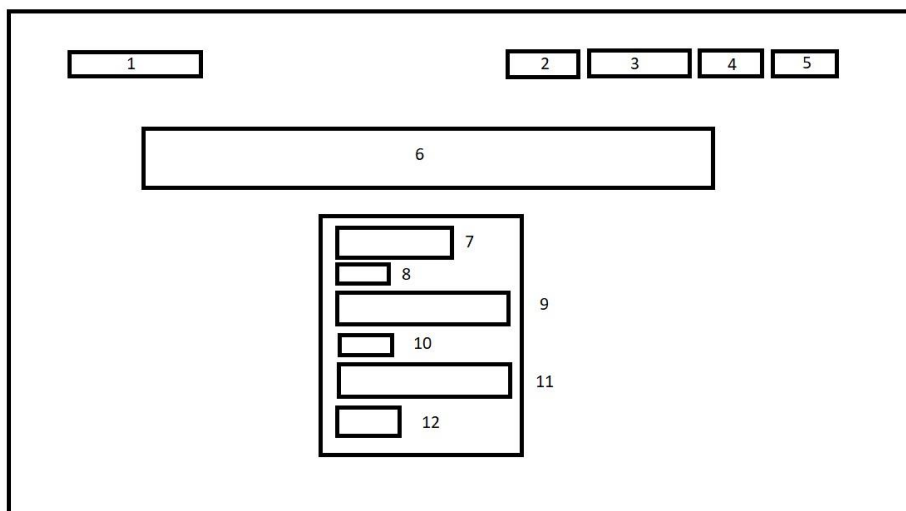
No	Nama Objek	Tipe	Keterangan
1	Nama Website	Text	Nama website bertuliskan Marketplace Komputer.
2	Home	Navigation bar	Dapat memindahkan User ke halaman panduan.
3	Kategori Komponen	Navigation bar	Dapat memindahkan User ke halaman kategori komponen.
4	Keranjang	Navigation bar	Dapat memindahkan User ke halaman keranjang.
5	Login	Navigation bar	Dapat memindahkan User ke halaman Login.
6	Selamat Datang	Text	Tulisan mengatakan selamat datang di website kami.
7	Judul Form	Text	Tulisan mengatakan User Login.
8	Nama Masukkan email	Text	Tulisan email.



9	Email	Input Text	Tempat User memasukkan email.
10	Nama password	Masukkan Text	Tulisan Password.
11	Password	Input	Tempat User memasukkan password.
12	Tombol Login	Button	Data (email dan password) akan dikirimkan ke sistem untuk diverifikasi akun.

#### 5.1.5.2 Perancangan Antarmuka Login Toko

Dibawah ini adalah rancangan dari halaman *login* untuk toko komputer sehingga user dapat masuk ke dalam sistem dan berubah menjadi aktor toko komputer yang disajikan pada gambar 5.8.



Gambar 5.8 Perancangan halaman antarmuka *login* toko

Pada tahap perancangan antarmuka halaman login user didapati menghasilkan 13 objek, yang keterangan dan tipe dari objek yang disajikan pada tabel 5.5.

Tabel 5.5 Objek antarmuka *login* toko

No	Nama Objek	Tipe	Keterangan
1	Nama Website	Text	Nama website bertuliskan Marketplace Komputer.
2	Home	Navigation bar	Dapat memindahkan User ke halaman panduan.

3	Kategori Komponen	<i>Navigation bar</i>	Dapat memindahkan <i>User</i> ke halaman kategori komponen.
4	Keranjang	<i>Navigation bar</i>	Dapat memindahkan <i>User</i> ke halaman keranjang.
5	<i>Login</i>	<i>Navigation bar</i>	Dapat memindahkan <i>User</i> ke halaman <i>Login</i> .
6	Selamat Datang	<i>Text</i>	Tulisan mengatakan selamat datang di website kami.
7	Judul <i>Form</i>	<i>Text</i>	Tulisan mengatakan <i>User Login</i> .
8	Nama Masukkan email	<i>Text</i>	Tulisan email.
9	<i>Email</i>	<i>Input Text</i>	Tempat <i>User</i> memasukkan email toko.
10	Nama Masukkan <i>password</i>	<i>Text</i>	Tulisan <i>Password</i> .
11	<i>Password</i>	<i>Input</i>	Tempat <i>User</i> memasukkan <i>password</i> toko.
12	Tombol <i>Login</i>	<i>Button</i>	Data (email dan password) akan dikirimkan ke sistem untuk diverifikasi akun.

### 5.1.5.3 Perancangan Antarmuka *Register User*

Dibawah ini adalah rancangan dari halaman *register* untuk *user* sehingga *user* dapat mengisi formulir untuk menjadi aktor *Member* yang disajikan pada gambar 5.9.

The diagram shows a registration form layout with the following components:

- 1: Input field for Username
- 2: Input field for Email
- 3: Input field for Password
- 4: Input field for Confirm Password
- 5: Input field for Phone Number
- 6: Large rectangular area for a profile picture or logo
- 7: Input field for First Name
- 8: Input field for Last Name
- 9: Input field for Address
- 10: Input field for City
- 11: Input field for Province
- 12: Input field for Zip Code
- 13: Input field for Registration Date



**Gambar 5 9 Rancangan halaman antarmuka *register user***

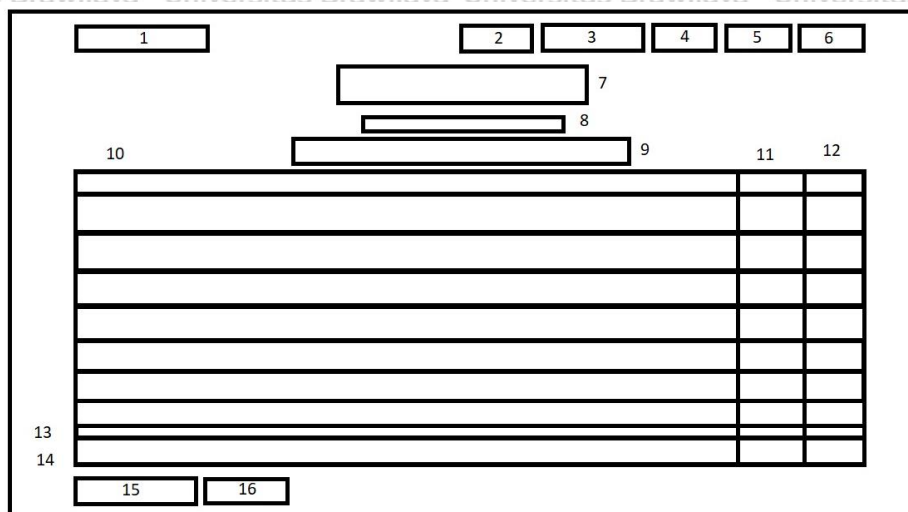
Pada tahap perancangan antarmuka halaman *register user* didapati menghasilkan 13 Objek, untuk keterangan dan tipe dari objek tersebut yang disajikan pada tabel 5.6.

**Tabel 5.6 Objek antarmuka *register user***

No	Nama Objek	Tipe	Keterangan
1	Nama Website	Text	Nama website bertuliskan Marketplace Komputer.
2	Home	Navigation bar	Dapat memindahkan User ke halaman panduan.
3	Kategori Komponen	Navigation bar	Dapat memindahkan User ke halaman kategori komponen.
4	Keranjang	Navigation bar	Dapat memindahkan User ke halaman keranjang.
5	Login	Navigation bar	Dapat memindahkan User ke halaman Login.
6	Selamat Datang	Text	Tulisan mengatakan selamat datang di website kami.
7	Judul Form	Text	Tulisan mengatakan User Login.
8	Nama Masukkan email	Text	Tulisan email.
9	Email	Input Text	Tempat User memasukkan email toko.
10	Nama Masukkan password	Text	Tulisan Password.
11	Password	Input	Tempat User memasukkan password toko.
12	Tombol Register	Button	Data (nama, email, password, dan konfirmasi password) akan dimasukkan ke database untuk pembuatan akun member.
13	Sudah punya akun	Link	Dapat memindahkan User ke halaman Login.

#### 5.1.5.4 Perancangan Antarmuka Halaman Simulasi

Di bawah ini adalah rancangan dari halaman simulasi untuk *member* dapat memilih komponen komputer untuk merakit satu komputer seutuhnya, yang disajikan pada gambar 5.10.



**Gambar 5.10 Rancangan halaman antarmuka simulasi**

Pada tahap perancangan antarmuka halaman *simulasi* didapati menghasilkan 16 Objek, untuk keterangan dan tipe dari objek tersebut yang disajikan pada tabel 5.7.

**Tabel 5.7 Objek antarmuka simulasi**

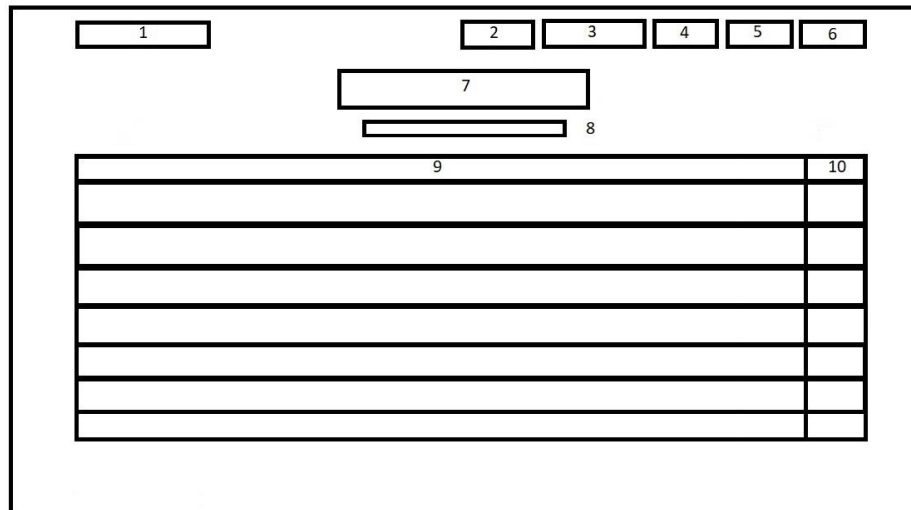
No	Nama Objek	Tipe	Keterangan
1	Nama Website	Text	Nama website bertuliskan Marketplace Komputer.
2	Home	Navigation bar	Dapat memindahkan <i>User</i> ke halaman panduan.
3	Kategori Komponen	Navigation bar	Dapat memindahkan <i>User</i> ke halaman kategori komponen.
4	Transaksi	Navigation bar	Dapat memindahkan <i>Member</i> ke halaman transaksi.
5	Keranjang	Navigation bar	Dapat memindahkan <i>User</i> ke halaman keranjang.
6	Logout	Navigation bar	Dapat mengeluarkan <i>Member</i> dari sistem untuk menjadi <i>User</i> .
7	Judul halaman	Text	Menampilkan tulisan sedang di halaman apa member saat ini.



8	Judul Div	Text	Menampilkan tulisan pemilihan komponen.
9	Penjelasan Div	Text	Menampilkan tulisan yang menjelaskan pemilihan komponen berdasarkan kompatibilitas.
10	Tabel komponen	Table	Menampilkan informasi komponen komputer.
11	Tabel aksi	Button	Menampilkan halaman daftar komponen komputer sesuai dengan kategori yang diinginkan.
12	Tabel buy	Input number dan button	Tempat User memasukkan jumlah produk yang ingin dibeli peserta memasukkan data yang telah diisikan oleh User ke keranjang.
13	Prediksi konsumsi listrik komputer	Text	Menampilkan angka prediksi konsumsi daya listrik komputer berdasarkan komponen komputer yang telah dipilih.
14	Total harga	Text	Menampilkan angka total dari harga komponen komputer yang berada pada halaman simulasi.
15	Cek power supply	Button	Mengambil data daya listrik pada komponen komputer yang berada pada halaman simulasi dan mengecek apakah power supply komputer dapat mengakomodir daya listrik komponen komputer.
16	Clear Simulasi	Button	Untuk membersihkan halaman simulasi

#### 5.1.5.5 Perancangan Antarmuka Halaman Daftar Produk (*Processor*)

Di bawah ini adalah rancangan dari halaman daftar produk untuk menampilkan katalog halaman processor, yang disajikan pada gambar 5.11.



**Gambar 5.11 Rancangan halaman antarmuka daftar produk**

Pada tahap perancangan antarmuka halaman daftar produk (*processor*) didapati menghasilkan 10 objek, untuk keterangan dan tipe dari objek tersebut yang disajikan pada tabel 5.8.

**Tabel 5.8 Objek antarmuka daftar produk (*processor*)**

No	Nama Objek	Tipe	Keterangan
1	Nama Website	Text	Nama website bertuliskan Marketplace Komputer.
2	Home	Navigation bar	Dapat memindahkan <i>User</i> ke halaman panduan.
3	Kategori Komponen	Navigation bar	Dapat memindahkan <i>User</i> ke halaman kategori komponen.
4	Transaksi	Navigation bar	Dapat memindahkan <i>Member</i> ke halaman transaksi.
5	Keranjang	Navigation bar	Dapat memindahkan <i>User</i> ke halaman keranjang.
6	Logout	Navigation bar	Dapat mengeluarkan <i>Member</i> dari sistem untuk menjadi <i>User</i> .
7	Judul halaman	Text	Menampilkan tulisan sedang di halaman apa member saat ini.
8	Judul Div	Text	Menampilkan tulisan pemilihan komponen.
9	Tabel komponen	Table	Menampilkan informasi komponen komputer.



10	Tabel aksi	Button	Menampilkan halaman daftar komponen komputer sesuai dengan kategori yang diinginkan.
----	------------	--------	--

#### 5.1.5.6 Perancangan Antarmuka Halaman Transaksi

Di bawah ini adalah rancangan dari halaman transaksi untuk *member* agar dapat melihat transaksi yang sedang terjadi dan sudah terjadi, yang disajikan pada gambar 5.12.

The diagram shows a wireframe for a transaction page. It includes a header area with boxes 1 through 6, a large central box 7, a box 8, and a table structure with boxes 9 through 11. The table has three rows and two columns, with the last column containing three small input boxes.

**Gambar 5.12 Rancangan halaman antarmuka transaksi**

Pada tahap perancangan antarmuka halaman transaksi didapati menghasilkan 11 objek, untuk keterangan dan tipe dari objek tersebut yang disajikan pada tabel 5.9.

**Tabel 5.9 Objek antarmuka transaksi**

No	Nama Objek	Tipe	Keterangan
1	Nama Website	Text	Nama website bertuliskan Marketplace Komputer.
2	Home	Navigation bar	Dapat memindahkan User ke halaman panduan.
3	Kategori Komponen	Navigation bar	Dapat memindahkan User ke halaman kategori komponen.
4	Transaksi	Navigation bar	Dapat memindahkan Member ke halaman transaksi.
5	Keranjang	Navigation bar	Dapat memindahkan User ke halaman keranjang.

6	Logout	Navigation bar	Dapat mengeluarkan <i>Member</i> dari sistem untuk menjadi <i>User</i> .
7	Judul halaman	Text	Menampilkan tulisan sedang di halaman apa <i>member</i> saat ini.
8	Navigasi transaksi	TextBreadcrumb	Menjelaskan lokasi <i>Member</i> saat itu.
9	Judul Div	Text	Menampilkan tulisan riwayat transaksi.
10	Tabel informasi transaksi	Tabel	Menampilkan informasi mengenai transaksi <i>Member</i> .
11	Tabel Aksi	Button	Dapat memindahkan <i>Member</i> ke halaman detail transaksi

#### 5.1.5.7 Perancangan Antarmuka Halaman Detail Pesanan

Di bawah ini adalah rancangan dari halaman detail pesanan untuk *member* agar dapat melihat detail transaksi yang dipilih dan sudah terjadi, yang disajikan pada gambar 5.13.



**Gambar 5.13 Rancangan halaman antarmuka detail pesanan**

Pada tahap perancangan antarmuka halaman detail pesanan didapati menghasilkan 17 objek, untuk keterangan dan tipe dari objek tersebut yang disajikan pada tabel 5.10.

**Tabel 5.10 Objek antarmuka detail pesanan**

No	Nama Objek	Tipe	Keterangan
----	------------	------	------------



1	Nama website	Text	Nama website bertuliskan Marketplace Komputer.
2	Home	Navigation bar	Dapat memindahkan <i>User</i> ke halaman panduan.
3	Kategori komponen	Navigation bar	Dapat memindahkan <i>User</i> ke halaman kategori komponen.
4	Transaksi	Navigation bar	Dapat memindahkan <i>Member</i> ke halaman transaksi.
5	Keranjang	Navigation bar	Dapat memindahkan <i>User</i> ke halaman keranjang.
6	Logout	Navigation bar	Dapat mengeluarkan <i>Member</i> dari sistem untuk menjadi <i>User</i> .
7	Judul halaman	Text	Menampilkan tulisan sedang di halaman apa member saat ini.
8	Navigasi transaksi	Breadcrumb	Menjelaskan lokasi <i>Member</i> saat itu.
9	Informasi transaksi	Text	Menampilkan tulisan apakah transaksi dalam status belum dibayar dan sudah lunas beserta tambahan informasi mengenai nomor telepon toko.
9	Judul Div detail Transaksi	Text	Menampilkan tulisan riwayat transaksi.
10	Judul Div detail transaksi	Text	Menampilkan tulisan detail transaksi.
11	Informasi tanggal pesan	Text	Menampilkan tulisan tanggal pemesanan barang.
12	Tabel informasi transaksi	Tabel	Menampilkan informasi barang yang dibeli oleh <i>Member</i> .
13	Tabel harga	Tabel	Menampilkan total harga barang serta biaya admin, dan total yang harus ditransferkan oleh <i>Member</i> .
14	Judul Div detail pengiriman	Text	Menampilkan tulisan detail pengiriman

15	Informasi pengiriman barang	Text	Menampilkan informasi mengenai barang pengiriman barang beserta nomor resi pengiriman.
16	Konfirmasi barang telah tiba	Button	Sebuah tombol untuk mengubah informasi transaksi menjadi sudah selesai.
17	Informasi Pengiriman	Text	Menampilkan informasi mengenai alamat pengiriman secara lengkap.

#### 5.1.5.8 Perancangan Antarmuka Halaman Keranjang

Di bawah ini adalah rancangan dari halaman keranjang untuk *member* agar dapat melihat komponen komputer apa saja yang sudah masuk ke dalam keranjang, yang disajikan pada gambar 5.14.

**Gambar 5.14 Rancangan halaman antarmuka keranjang**

Pada tahap perancangan antarmuka halaman transaksi didapati menghasilkan 15 objek, untuk keterangan dan tipe dari objek tersebut yang disajikan pada tabel 5.11.

**Tabel 5.11 Objek antarmuka keranjang**

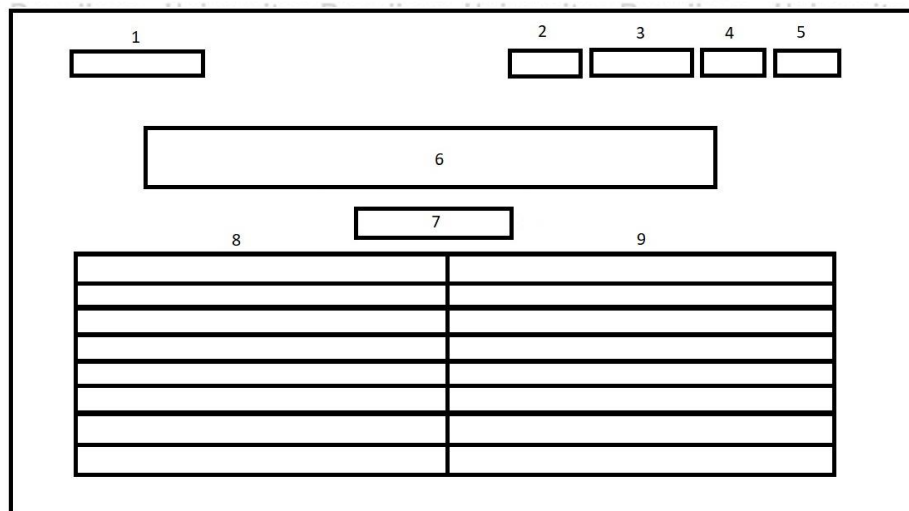
No	Nama Objek	Tipe	Keterangan
1	Nama website	Text	Nama website bertuliskan Marketplace Komputer.
2	Home	Navigation bar	Dapat memindahkan <i>User</i> ke halaman panduan.



3	Kategori komponen	<i>Navigation bar</i>	Dapat memindahkan <i>User</i> ke halaman kategori komponen.
4	Transaksi	<i>Navigation bar</i>	Dapat memindahkan <i>Member</i> ke halaman transaksi.
5	Keranjang	<i>Navigation bar</i>	Dapat memindahkan <i>User</i> ke halaman keranjang.
6	<i>Logout</i>	<i>Navigation bar</i>	Dapat mengeluarkan <i>Member</i> dari sistem untuk menjadi <i>User</i> .
7	Judul halaman	<i>Text</i>	Menampilkan tulisan sedang di halaman apa member saat ini.
8	Judul div	<i>Text</i>	Tulisan mengatakan daftar keranjang.
9	Detail barang	<i>Table</i>	Menampilkan nama komponen komputer, harga barang, jumlah barang, dan total harga.
10	Tabel aksi	<i>Table</i>	Tabel untuk mengakomodir tombol.
11	Hapus	<i>Button</i>	Untuk menghapus barang yang telah.
12	Bayar	<i>Button</i>	Untuk beralih ke halaman pembayaran dengan isi barang yang berada di keranjang.
13	Tanggal	<i>Text</i>	Menampilkan tanggal pembuatan keranjang.

#### 5.1.5.9 Perancangan Antarmuka Halaman Home Toko

Di bawah ini adalah rancangan dari halaman utama untuk toko dapat melihat data identitas dari toko, yang disajikan pada gambar 5.15.



**Gambar 5.15 Rancangan halaman antarmuka *home* toko**

Pada tahap perancangan antarmuka halaman *home* toko didapati menghasilkan 9 objek, untuk keterangan dan tipe dari objek tersebut yang disajikan pada tabel 5.12.

**Tabel 5.12 Objek antarmuka *home* toko**

No	Nama Objek	Tipe	Keterangan
1	Nama website	<i>Text</i>	Nama website bertuliskan Marketplace Komputer.
2	<i>Home</i>	<i>Navigation bar</i>	Dapat memindahkan <i>User</i> ke halaman panduan.
3	Kategori komponen	<i>Navigation bar</i>	Dapat memindahkan <i>User</i> ke halaman kategori komponen.
4	Pesanan	<i>Navigation bar</i>	Dapat memindahkan Toko ke halaman pesanan.
5	<i>Logout</i>	<i>Navigation bar</i>	Dapat mengeluarkan Toko dari sistem untuk menjadi <i>User</i> .
6	Judul halaman	<i>Text</i>	Menampilkan Tulisan mengatakan selamat datang toko.
7	Data account toko	<i>Text</i>	Menampilkan tulisan data account toko.
8	Tabel Head Toko	<i>Table Head</i>	Menampilkan tulisan nama toko, email, alamat, nomor hp, rekening bank, nomor rekening, atas nama rekening, tanggal pembuatan akun toko.



9	Tabel Data Toko	Table Data	Menampilkan data dari nama toko, email, alamat, nomor hp, nomor rekening, atas nama rekening, tanggal pembuatan akun toko.
---	-----------------	------------	--

#### 5.1.5.10 Perancangan antarmuka Halaman Daftar Transaksi

Di bawah ini adalah rancangan dari halaman daftar transaksi untuk toko dapat melihat data daftar transaksi pada toko yang disajikan pada gambar 5.16.

**Gambar 5.16 Rancangan halaman antarmuka daftar transaksi**

Pada tahap perancangan antarmuka halaman daftar transaksi didapati menghasilkan 10 objek, untuk keterangan dan tipe dari objek tersebut yang disajikan pada tabel 5.11.

**Tabel 5.13 Objek antarmuka daftar transaksi**

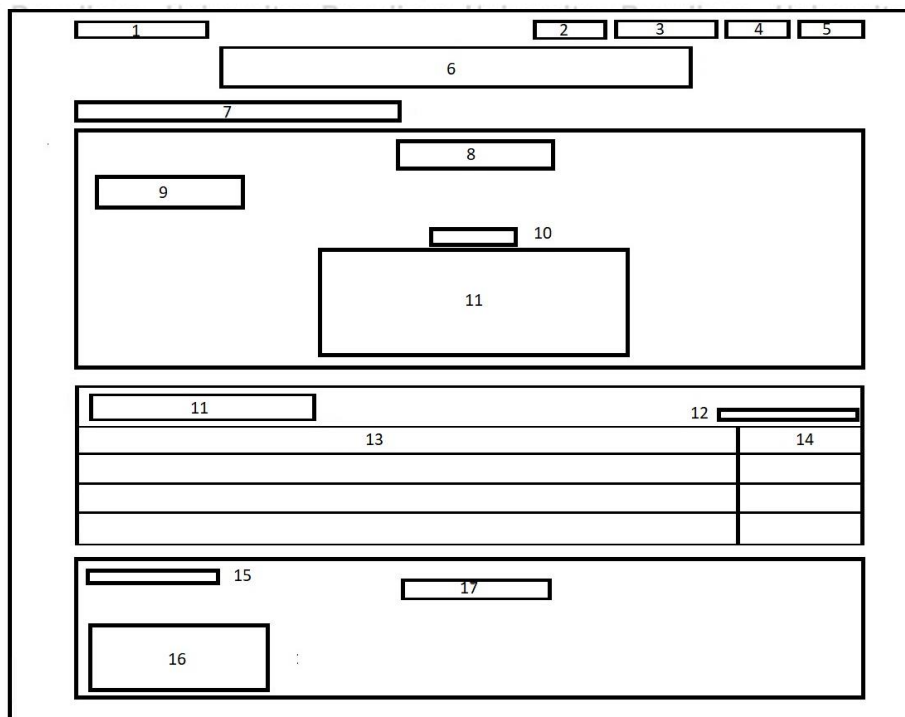
No	Nama Objek	Tipe	Keterangan
1	Nama website	Text	Nama website bertuliskan Marketplace Komputer.
2	Home	Navigation bar	Dapat memindahkan User ke halaman panduan.
3	Kategori komponen	Navigation bar	Dapat memindahkan User ke halaman kategori komponen.
4	Pesanan	Navigation bar	Dapat memindahkan Toko ke halaman pesanan.

5	Logout	Navigation bar	Dapat mengeluarkan Toko dari sistem untuk menjadi User.
6	Judul halaman	Text	Menampilkan Tulisan mengatakan selamat datang toko.
7	Navigasi pesanan	Breadcrumb	Menjelaskan lokasi Toko saat itu
8	Daftar pesanan toko	Text	Menampilkan tulisan daftar pesanan toko.
9	Daftar tabel data pesanan toko	Table	Menampilkan data tanggal pesanan, kode transaksi pesanan, menampilkan status pesanan, menampilkan jumlah harga.
10	Detail	Button	Menampilkan data dari nama toko, email, alamat, nomor hp, nomor rekening, atas nama rekening, tanggal pembuatan akun toko.

#### 5.1.5.11 Perancangan Antarmuka Halaman Detail Pesanan

Di bawah ini adalah rancangan dari halaman detail pesanan untuk toko, agar dapat melihat detail pesanan dari transaksi yang dipilih, yang disajikan pada gambar 5.17.





**Gambar 5.17 Rancangan halaman antarmuka detail pesanan**

Pada tahap perancangan antarmuka halaman detail pesanan didapati menghasilkan 16 objek, untuk keterangan dan tipe dari objek tersebut yang disajikan pada tabel 5.14.

**Tabel 5.14 Objek antarmuka detail pesanan**

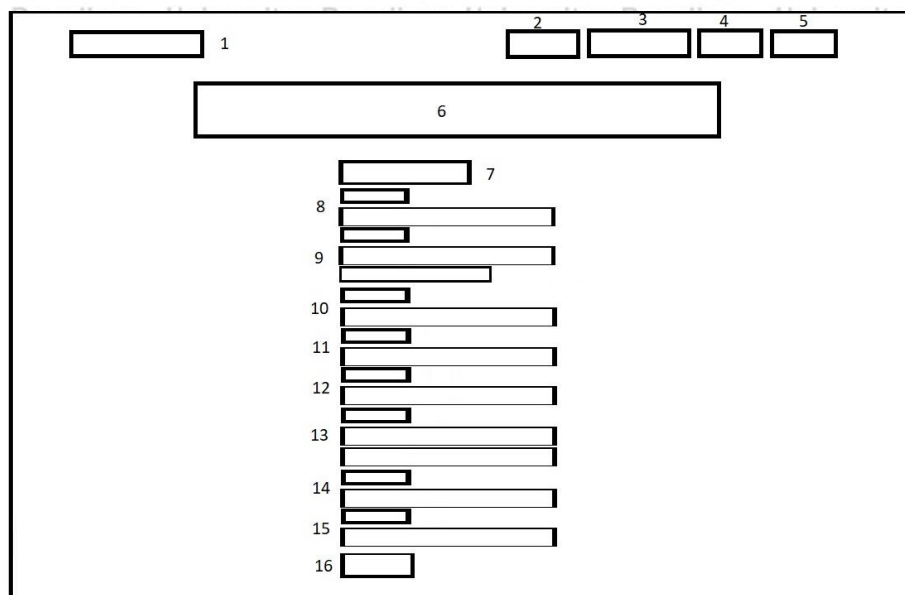
No	Nama Objek	Tipe	Keterangan
1	Nama website	Text	Nama website bertuliskan Marketplace Komputer.
2	Home	Navigation bar	Dapat memindahkan User ke halaman panduan.
3	Kategori komponen	Navigation bar	Dapat memindahkan User ke halaman kategori komponen.
4	Pesanan	Navigation bar	Dapat memindahkan Toko ke halaman pesanan.
5	Logout	Navigation bar	Dapat mengeluarkan Toko dari sistem untuk menjadi User.
6	Judul halaman	Text	Menampilkan Tulisan mengatakan selamat datang toko.
7	Navigasi pesanan	Breadcrumb	Menjelaskan lokasi Toko saat itu.

8	Informasi transaksi	Text	Menampilkan tulisan apakah pemesanan belum dibayar dan sudah lunas beserta tambahan informasi mengenai nomor telepon toko.
9	Judul Div detail pesanan	Text	Menampilkan tulisan detail transaksi.
10	Informasi tanggal pesan	Text	Menampilkan tulisan tanggal pemesanan barang dibuat.
11	Detail pesanan	Text	Menampilkan tulisan detail pesanan.
12	Tabel informasi transaksi	Tabel	Menampilkan informasi barang yang dibeli oleh member.
13	Tabel harga	Tabel	Menampilkan total harga barang serta biaya admin, dan total yang harus ditransfer oleh member.
14	Judul Div detail transaksi	Text	Menampilkan tulisan detail pengiriman.
15	Informasi pengiriman barang	Text	Menampilkan informasi mengenai barang pengiriman barang beserta nomor resi pengiriman.
16	Informasi alamat pengiriman	Text	Menampilkan informasi mengenai alamat pengiriman berupa nama penerima, nomor telfon, nama penerima, alamat, provinsi, kota, kecamatan, kelurahan, kode pos.
17	Status pengiriman	Text	Menampilkan informasi status pengiriman.

#### 5.1.5.12 Perancangan antarmuka Halaman Tambah Produk (*Processor*)

Di bawah ini adalah rancangan dari halaman untuk tambah produk processor agar toko dapat menambahkan barang jualan kategori processor, yang disajikan pada gambar 5.18.





**Gambar 5.18 Rancangan halaman antarmuka tambah produk (*Processor*)**

Pada tahap perancangan antarmuka halaman tambah produk *processor* didapati menghasilkan 16 objek, untuk keterangan dan tipe dari objek tersebut yang disajikan pada tabel 5.15.

**Tabel 5.15 Objek antarmuka tambah produk (*Processor*)**

No	Nama Objek	Tipe	Keterangan
1	Nama website	<i>Text</i>	Nama website bertuliskan Marketplace Komputer.
2	<i>Home</i>	<i>Navigation bar</i>	Dapat memindahkan toko komputer ke halaman panduan.
3	Kategori komponen	<i>Navigation bar</i>	Dapat memindahkan toko komputer ke halaman kategori komponen.
4	Pesanan	<i>Navigation bar</i>	Dapat memindahkan Toko komputer ke halaman pesanan.
5	<i>Logout</i>	<i>Navigation bar</i>	Dapat mengeluarkan Toko komputer dari sistem untuk menjadi User.
6	Judul halaman	<i>Text</i>	Menampilkan Tulisan mengatakan selamat datang toko.
7	Judul form	<i>Text</i>	Tulisan mengatakan <i>form</i> produk <i>processor</i> .

8	Nama processor	Input text	Tempat toko memasukkan data nama processor.
9	Soket processor	Input text	Tempat toko memasukkan data soket processor.
11	Tdp processor	Input text	Tempat toko memasukkan data tdp processor.
12	Boost clock processor	Input text	Tempat toko memasukkan data boost clock processor.
13	Integrated graphic	Input radio	Tempat toko memasukkan data integrated graphic processor.
14	Harga processor	Input number	Tempat toko memasukkan data harga processor.
15	Stok processor	Input number	Tempat toko memasukkan data stok processor.
16	Tambah	button	Sistem akan menyimpan masukkan dari toko.

### 5.1.5.13 Perancangan antarmuka Halaman Edit Produk (*Processor*)

Di bawah ini adalah rancangan dari halaman untuk mengedit produk processor agar toko dapat mengedit barang jualan kategori processor, yang disajikan pada gambar 5.19.

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16



### Gambar 5.19 Rancangan halaman antarmuka tambah prdouk (*Processor*)

Pada tahap perancangan antarmuka halaman tambah produk *processor* didapati menghasilkan 16 objek, untuk keterangan dan tipe dari objek tersebut yang disajikan tabel 5.16.

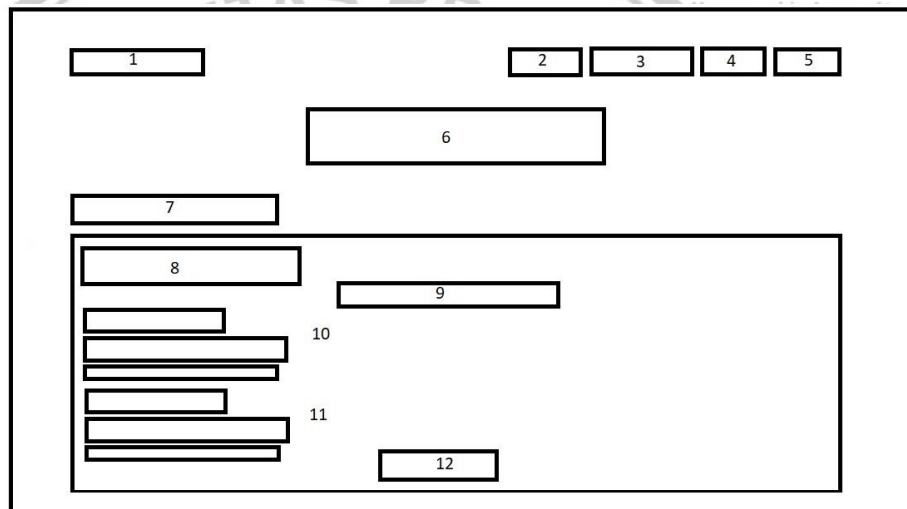
**Tabel 5.16 Objek antarmuka edit produk (*Processor*)**

No	Nama Objek	Tipe	Keterangan
1	Nama website	Text	Nama website bertuliskan Marketplace Komputer.
2	Home	Navigation bar	Dapat memindahkan User ke halaman panduan.
3	Kategori komponen	Navigation bar	Dapat memindahkan User ke halaman kategori komponen.
4	Pesanan	Navigation bar	Dapat memindahkan Toko komputer ke halaman pesanan.
5	Logout	Navigation bar	Dapat mengeluarkan Toko komputer dari sistem untuk menjadi User.
6	Judul halaman	Text	Menampilkan Tulisan mengatakan selamat datang toko.
7	Judul form	Text	Tulisan mengatakan form produk processor.
8	Nama processor	Input text	Tempat toko komputer memasukkan data nama processor.
9	Soket processor	Input text	Tempat toko komputer memasukkan data soket processor.
11	Tdp processor	Input text	Tempat toko memasukkan data tdp processor.
12	Boost clock processor	Input text	Tempat toko komputer memasukkan data boost clock processor.
13	Integrated graphic	Input radio	Tempat toko komputer memasukkan data integrated graphic processor.

14	Harga processor	<i>Input number</i>	Tempat memasukkan data harga processor.
15	Stok processor	<i>Input number</i>	Tempat memasukkan data stok processor.
16	Edit	<i>button</i>	Sistem akan menyimpan masukan dari toko komputer.

#### 5.1.5.14 Perancangan antarmuka Halaman Isi Pengiriman

Di bawah ini adalah rancangan dari halaman untuk isi pengiriman toko komputer sehingga dapat mengisi formulir berupa nama kurir dan nomor resi, yang disajikan pada gambar 5.20.



Gambar 5.20 Rancangan halaman antarmuka isi pengiriman

Pada tahap perancangan antarmuka halaman isi pengiriman didapati menghasilkan 12 objek, untuk keterangan dan tipe dari objek tersebut yang disajikan pada tabel 5.17.

Tabel 5.17 Objek antarmuka isi pengiriman

No	Nama Objek	Tipe	Keterangan
1	Nama website	<i>Text</i>	Nama website bertuliskan Marketplace Komputer.
2	<i>Home</i>	<i>Navigation bar</i>	Dapat memindahkan User ke halaman panduan.
3	Kategori komponen	<i>Navigation bar</i>	Dapat memindahkan User ke halaman kategori komponen.



4	Pesanan	<i>Navigation bar</i>	Dapat memindahkan Toko ke halaman pesanan.
5	<i>Logout</i>	<i>Navigation bar</i>	Dapat mengeluarkan Toko dari sistem untuk menjadi <i>User</i> .
6	Judul halaman	<i>Text</i>	Menampilkan Tulisan mengatakan selamat datang toko.
7	Navigasi transaksi	<i>Breadcrumb</i>	Menjelaskan lokasi toko saat itu.
8	Data kurir	<i>Text</i>	Menampilkan tulisan data kurir pengiriman sebagai judul dari div.
9	Pengingat	<i>Text</i>	Menampilkan tulisan “Pastikan data yang dimasukkan adalah benar!”
11	Nomor resi	<i>Input text</i>	Tempat toko memasukkan data nama kurir.
12	Simpan	<i>Button</i>	Menyimpan data (kurir dan nomor resi) ke dalam sistem.

## 5.2 Implementasi Sistem

Apabila proses perancangan telah selesai dibuat, dapat dilakukannya implementasi sitem. Proses berlangsungnya implementasi didasari oleh hasil dari proses analisis kebutuhan serta proses perancangan. Tahap ini menjelaskan gambaran mengenai implementasi antarmuka, implementasi data, serta implementasi kode.

### 5.2.1 Implementasi Antarmuka

Hasil akhir dari perancangan antarmuka adalah implementasi antarmuka. Tahap ini memberikan 14 gambaran implementasi antarmuka, yaitu implementasi antarmuka pada halaman *login* untuk *user*, halaman *login* untuk toko komputer, halaman *register* untuk *user*, halaman simulasi, halaman daftar produk, halaman transaksi, halaman detail pesanan, halaman keranjang, halaman *home* toko, halaman daftar transaksi, halaman detail transaksi, halaman tambah produk (*Processor*), halaman edit transaksi (*Processor*), dan halaman tambah pengiriman.

### 5.2.1.1 Implementasi Antarmuka Login User

**Gambar 5.21 Implementasi antarmuka login user**

Gambar 5.21 merupakan hasil aplikasi antarmuka *login* untuk *user*. Pada halaman ini terdapat sebuah formulir yang berisi alamat *email* dan *password*. Halaman ini dibuat agar *user* dapat masuk ke dalam sistem sebagai *member*.

### 5.2.1.2 Implementasi Antarmuka Login Toko Komputer

**Gambar 5.22 Implementasi antarmuka login toko komputer**

Gambar 5.22 merupakan hasil aplikasi antarmuka *login* untuk toko komputer. Pada halaman ini terdapat sebuah formulir yang berisi alamat *email* dan *password*. Halaman ini dibuat untuk *user* dapat masuk ke dalam sistem sebagai toko komputer.



### 5.2.1.3 Implementasi Antarmuka Register User

Marketplace Komputer

Home Kategori Komponen Keranjang Login

Selamat Datang Di Website Kami

User Register

Name  
Enter full name

Email  
Enter email address

Password  
Enter password

Confirm Password  
Enter confirm password

Register

[I already have an account](#)

**Gambar 5.23 Implementasi antarmuka register user**

Gambar 5.23 merupakan hasil aplikasi antarmuka *register* untuk *user*. Pada halaman ini terdapat sebuah formulir yang berisi nama, *email*, *password*, dan konfirmasi *password*. Selain itu, terdapat tombol “Register” agar sistem dapat menyimpan data yang sudah dimasukkan oleh *user*. Halaman ini dibuat untuk *user* dapat masuk ke dalam sistem dengan menjadi member.

#### 5.2.1.4 Implementasi Antarmuka Halaman Simulasi

Marketplace Komputer Simulasi Kategori Komponen Transaksi Keranjang Logout

### Halaman Simulasi

#### Halaman Pemilihan Komponen

Dengan Memperhatikan Kompatibilitas Antara Processor, Motherboard, RAM  
Sekaligus Menghitung Prediksi Konsumsi Daya Listrik Komputer

Komponen	Nama	Harga	Aksi	Buy?
Processor		Rp.	+	<input type="text"/> Beli
Motherboard		Rp.	+	<input type="text"/> Beli
RAM		Rp.	+	<input type="text"/> Beli
VGA Card		Rp.	+	<input type="text"/> Beli
Storage		Rp.	+	<input type="text"/> Beli
Casing		Rp.	+	<input type="text"/> Beli
Power Supply		Rp.	+	<input type="text"/> Beli

Total Prediksi Watt Komputer 150 Watt

Total Harga Rp. 0

Cek Power Supply Komputer Clear Simulasi

Gambar 5.24 Implementasi antarmuka halaman simulasi

Gambar 5.24 merupakan hasil aplikasi antarmuka simulasi. Pada halaman ini terdapat sebuah tabel yang berisi komponen-komponen komputer, seperti *Processor*, *Motherboard*, *RAM*, *VGA Card*, *Storage*, *Casing*, dan *Power Supply*. Selain itu terdapat tombol “+” berwarna biru untuk sistem menampilkan halaman komponen yang telah dipilih. Terdapat juga kolom “Buy?” yang menampilkan formulir dengan tipe nomor untuk *member* memasukkan jumlah komponen yang ingin dibeli. Pada kolom tersebut juga terdapat tombol “Beli” yang akan ditekan oleh *member* untuk sistem memasukkan komponen ke dalam keranjang. Terdapat juga baris yang memberikan keterangan berapa total daya yang akan dihasilkan oleh komputer serta total harga yang harus dibayarkan oleh *member*. Pada bagian bawah halaman terdapat 2 tombol tambahan, yaitu tombol “Cek Power Supply Komputer” yang berwarna kuning untuk memastikan kompatibilitas komponen dan tombol “Clear Simulasi” untuk *member* jika ingin menghapus seluruh data yang ada pada halaman simulasi.



### 5.2.1.5 Implementasi Antarmuka Halaman Daftar Produk (Processor)

Marketplace Komputer Simulasi Kategori Komponen - History Transaksi Logout

## Halaman Katalog Kategori Processor

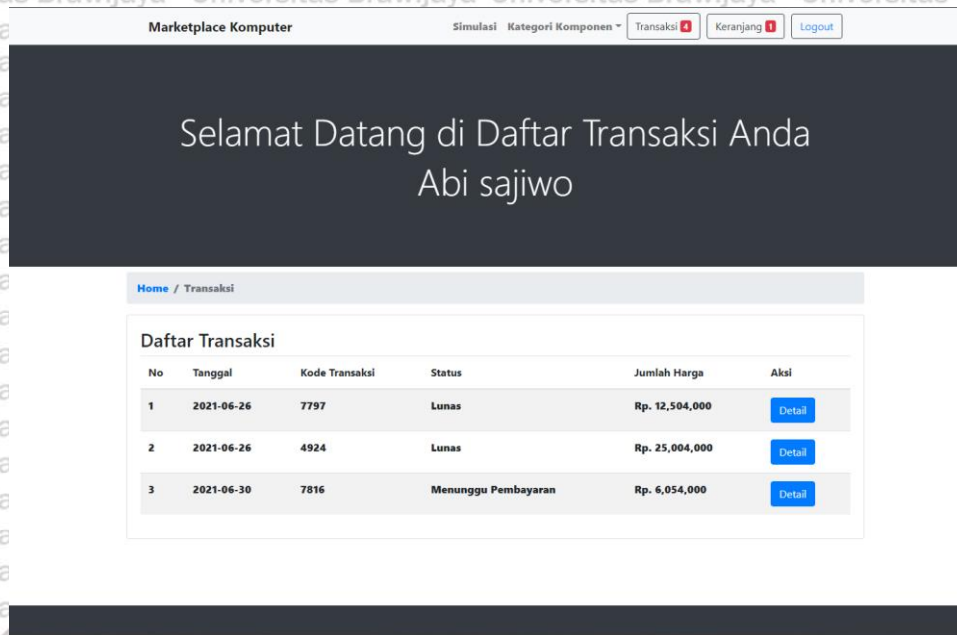
Daftar Produk Processor

No	Nama Barang	Soket	TDP	Core Clock processor	Integrated_graphic	Harga	Stok	Aksi
1	Intel Core i7 7800X	2066	140 watt	3.5 Ghz	0	Rp. 6050000	2	<input type="text"/> +
2	INTEL CORE i5 10400F	1200	65 watt	2.9 Ghz	1	Rp. 2575000	4	<input type="text"/> +
3	INTEL CORE i5 11600	1200	65 watt	2.8 Ghz	1	Rp. 4300000	10	<input type="text"/> +
4	Intel Core i5 8400	1151	65 watt	2.8 Ghz	1	Rp. 3400000	4	<input type="text"/> +
5	AMD RYZEN 5 2400G 4- Core	AM4	65 watt	3.6 Ghz	1	Rp. 1950000	3	<input type="text"/> +
6	Intel Core i7 8700	1151	65 watt	3.2 Ghz	1	Rp. 6125000	7	<input type="text"/> +

**Gambar 5.25 Implementasi antarmuka daftar produk**

Gambar 5.25 merupakan hasil aplikasi antarmuka daftar produk untuk salah satu komponen, yaitu *Processor*. Pada halaman ini terdapat sebuah tabel yang berisi data processor, yaitu nama barang, soket, TDP, *core clock processor*, *integrated\_graphic*, harga, dan stok barang. Selain itu, terdapat kolom "Aksi" yang menampilkan formulir dengan tipe nomor untuk *member* memasukkan jumlah *Processor* yang ingin dibeli. Pada kolom tersebut juga terdapat tombol "+" yang akan ditekan oleh *member* untuk sistem memasukkan komponen ke dalam keranjang.

### 5.2.1.6 Implementasi Antarmuka Halaman Transaksi

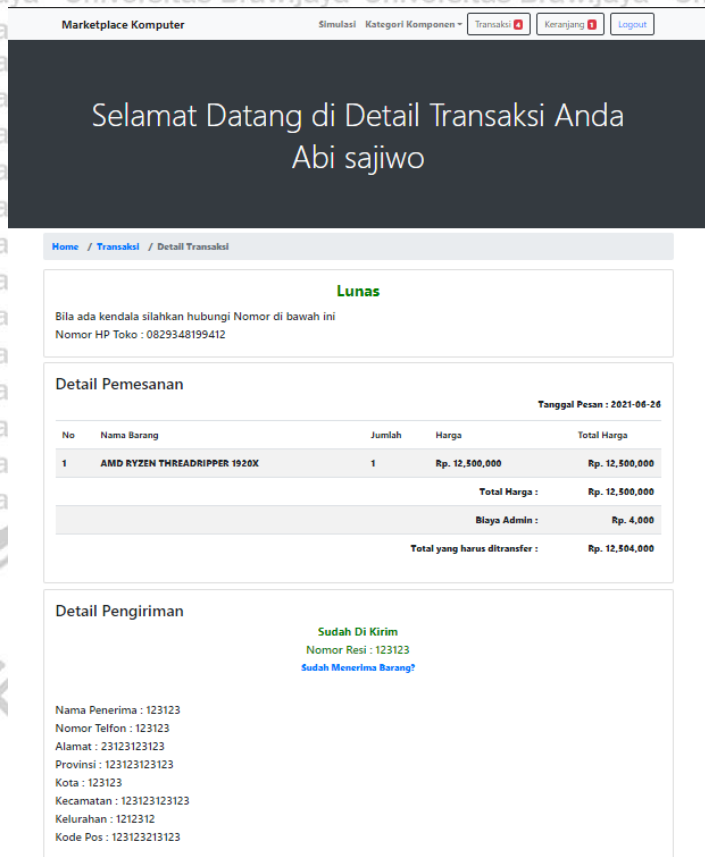


**Gambar 5.26 Implementasi antarmuka transaksi**

Gambar 5.26 merupakan hasil aplikasi antarmuka transaksi untuk *member*. Pada halaman ini terdapat sebuah *breadcrumb* yang menjadi petunjuk kepada *member* mengenai keberadaan dirinya saat itu. Selain itu terdapat sebuah tabel yang berisi daftar transaksi *member* yang berisi nomor, tanggal, kode transaksi, status, dan jumlah harga yang harus dibayarkan oleh *member*. Terdapat juga kolom “Aksi” yang menampilkan tombol “Detail” yang dapat *member* tekan agar sistem dapat menampilkan halaman detail pesanan.



### 5.2.1.7 Implementasi Antarmuka Halaman Detail Transaksi



**Gambar 5.27 Implementasi antarmuka detail transaksi (Processor)**

Gambar 5.27 merupakan hasil aplikasi antarmuka detail transaksi untuk *member*. Pada halaman ini terdapat *breadcrumb* yang menjadi petunjuk kepada *member* mengenai keberadaan dirinya saat itu. Selain itu terdapat kolom mengenai informasi transaksi yang berisi status pembayaran serta informasi tambahan, yaitu nomor telepon toko. Terdapat juga tabel mengenai detail pesanan yang berisi nomor, nama barang, jumlah, harga, dan total harga. Di akhir halaman, terdapat kolom detail pengiriman barang yang berisi mengenai status pesanan, nomor resi yang diberikan oleh toko, tombol "Sudah Menerima Barang" yang akan ditekan oleh *member* ketika barang sudah tiba, dan alamat *member*.

### 5.2.1.8 Implementasi Antarmuka Halaman Keranjang

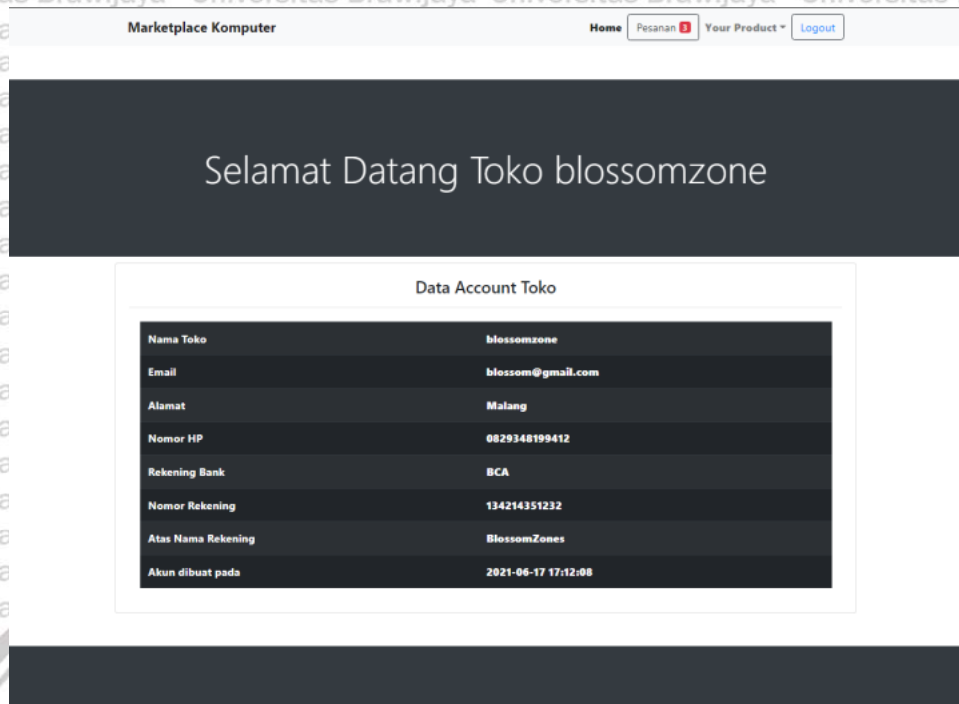


**Gambar 5.28 Implementasi antarmuka keranjang**

Gambar 5.28 merupakan hasil aplikasi antarmuka keranjang. Pada halaman ini terdapat sebuah tabel daftar barang yang berada di keranjang yang berisi nomor, nama barang, harga barang, jumlah barang, dan total harga. Selain itu terdapat juga kolom "Aksi" yang menampilkan tombol "X" untuk *member* tekan saat ingin menghapus barang dari keranjang serta tombol "Bayar" untuk *member* tekan ketika ingin melakukan pembayaran. Di akhir halaman terdapat tanggal pembuatan keranjang.



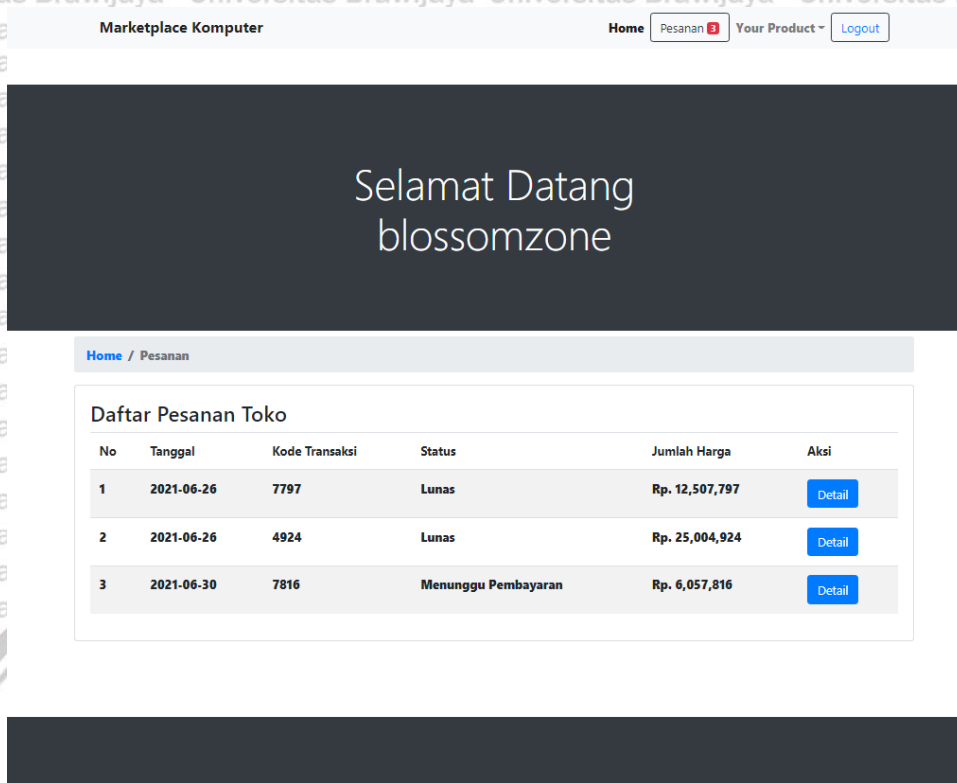
### 5.2.1.9 Implementasi Antarmuka Halaman *Home* Toko Komputer



**Gambar 5.29 Implementasi antarmuka *home* toko komputer**

Gambar 5.29 merupakan hasil aplikasi antarmuka *home* untuk toko komputer. Pada halaman ini terdapat sebuah pesan selamat datang kepada toko komputer. Selain itu terdapat sebuah tabel mengenai informasi toko seperti nama toko, alamat *email*, nomor *handphone*, jenis rekening bank, nomor rekening, atas nama siapa rekening bank tersebut, dan tanggal pembuatan akun.

### 5.2.1.10 Implementasi Antarmuka Halaman Daftar Transaksi



**Gambar 5.30 Implementasi antarmuka daftar transaksi**

Gambar 5.30 merupakan hasil aplikasi antarmuka datar transaksi untuk toko komputer. Pada halaman ini terdapat breadcrumb yang menjadi petunjuk kepada toko mengenai keberadaannya saat itu. Selain itu terdapat sebuah tabel daftar pesanan yang masuk ke toko dan berisi nomor, tanggal pesanan, kode transaksi, status, dan jumlah harga. Terdapat juga kolom "Aksi" yang menampilkan tombol "Detail" yang dapat toko tekan untuk sistem menampilkan halaman detail transaksi.



### 5.2.1.11 Implementasi Antarmuka Halaman Detail Transaksi

Marketplace Komputer

Nama
Pencarian
Your Product
login

Selamat Datang  
blossomzone

Home / Pesanan / Detail Pesanan

Menunggu Konfirmasi Pembayaran

Kode Transaksi : 7816

Nominal Rupiah : Rp. 6,054,000

Bukti Pembayaran dari Pembeli

Sudah Menerima Pembayaran?

Detail Pesanan

Tanggal Pesan : 2021-06-30

No	Nama Barang	Jumlah	Harga	Total Harga
1	Intel Core i7 7800X	1	Rp. 6,050,000	Rp. 6,050,000
Total Harga :				Rp. 6,050,000
Biaya Admin :				Rp. 4,000
Total yang harus ditransfer :				Rp. 6,054,000

Detail Pengiriman

Nama Penerima : 1231231

Nomor Telfon : 2312312

Nama Penerima : 1231231

Alamat : 312312

Provinsi : 31231231

Kota : 2312312

Kecamatan : 312312

Kelurahan : 3123123

Kode Pos : 12312313

Gambar 5.31 Implementasi antarmuka detail transaksi

Gambar 5.31 merupakan hasil aplikasi antarmuka detail transaksi untuk toko komputer. Pada halaman ini terdapat *breadcrumb* yang menjadi petunjuk kepada toko komputer mengenai keberadaannya saat itu. Selain itu terdapat kolom detail pembayaran yang berisi status pembayaran, kode transaksi, nominal

yang dibayarkan oleh *member*, foto bukti hasil pembayaran, dan tombol “Lunas” yang dapat ditekan oleh toko untuk mengubah status menjadi “Lunas”. Kemudian, terdapat sebuah tabel detail pesanan yang berisi nomor, nama barang, jumlah barang, harga barang, total harga yang harus dibayarkan oleh *member*, serta tanggal pesanan yang terletak di atas tabel. Pada akhir halaman terdapat kolom detail pengiriman barang yang berisikan alamat pembeli.

#### 5.2.1.12 Implementasi Antarmuka Halaman Tambah Produk (*Processor*)

The screenshot displays the 'Marketplace Komputer' website interface. At the top, there is a navigation bar with links for 'Home', 'Pesanan', 'Your Product', and 'Logout'. Below this, a large dark banner reads 'Halaman Tambah Produk Processor'. The main content area is titled 'Data Produk Processor' and contains a form with the following fields:

- Nama Processor:** A text input field with the placeholder 'Nama Processor'.
- Soket Processor:** A text input field with the placeholder 'Enter Soket Processor' and a note 'contoh: 1150 atau 1200 atau 1151'.
- TDP processor (watt):** A text input field with the placeholder 'Enter TDP processor'.
- Core Clock Processor (Ghz):** A text input field with the placeholder 'Masukkan core clock processor'.
- Boost Clock Processor (Ghz):** A text input field with the placeholder 'Masukkan Boost clock processor'.
- Integrated Graphic:** Two radio button options: 'Ada' and 'Tidak Ada'.
- Harga Processor:** A text input field with a 'Rp.' prefix and a placeholder.
- Stock Processor:** A text input field with the placeholder 'Masukkan jumlah stock'.
- Tambah Barang:** A blue button at the bottom of the form.

**Gambar 5.32 Implementasi antarmuka tambah produk (*Processor*)**

Gambar 5.32 merupakan hasil aplikasi antarmuka tambah produk untuk salah satu komponen komputer, yaitu *Processor*. Pada halaman ini terdapat sebuah formulir data produk *Processor* yang berisi nama, soket, TDP, *Core Clock*, *Boost Clock*, *integrated graphic*, harga, dan stok barang. Selain itu terdapat tombol “Tambah Barang” yang dapat toko tekan untuk sistem menyimpan data produk dan menampilkannya pada halaman daftar barang jualan.



### 5.2.1.13 Implementasi Antarmuka Halaman Edit Produk (*Processor*)

Marketplace Komputer

Home Pesanan 4 Your Product Logout

## Halaman Edit Produk Processor

Edit Data Produk Processor

**Nama Processor**

**Soket Processor**  
  
contoh: 1150 atau 1200 atau 1151

**TDP processor**

**Core Clock Processor (Ghz)**

**Boost Clock Processor (Ghz)**

**Integrated Graphic**  
☐ Ada  
☐ Tidak Ada

**Harga Processor**  
 Rp.

**Stock Processor**

[Update Data Processor](#)

**Gambar 5.33 Implementasi antarmuka edit produk (*Processor*)**

Gambar 5.33 merupakan hasil aplikasi antarmuka edit produk untuk salah satu komponen komputer, yaitu *Processor*. Pada halaman ini terdapat formulir untuk mengedit data produk yang berisi nama, soket, TDP, *Core Clock*, *Boost Clock*, *integrated graphic*, harga, dan stok barang. Selain itu terdapat tombol “Update Data Processor” yang dapat toko tekan untuk sistem menyimpan perubahan data produk dan menampilkannya pada halaman daftar barang jualan.

#### 5.2.1.14 Implementasi Antarmuka Halaman isi Pengiriman

The screenshot displays the Blossomzone Marketplace Komputer interface. At the top, there is a navigation bar with links for Home, Pesanan, Your Product, and Logout. Below this, a large black banner with white text reads "Selamat Datang blossomzone". Underneath the banner, a breadcrumb trail shows the path: Home / Transaksi / Detail Transaksi / Alamat Pengiriman. The main content area features a form titled "Data Kurir Pengiriman" with a sub-header "Pastikan data yang dimasukkan benar!". The form has two sections: "Kurir" with a text input field and a hint "Contoh : JNE / JNT / sicepat / Tiki", and "No Resi" with a text input field and a hint "Pastikan Nomor Resi yang Dimasukkan benar". A blue "Simpan" button is located at the bottom right of the form.

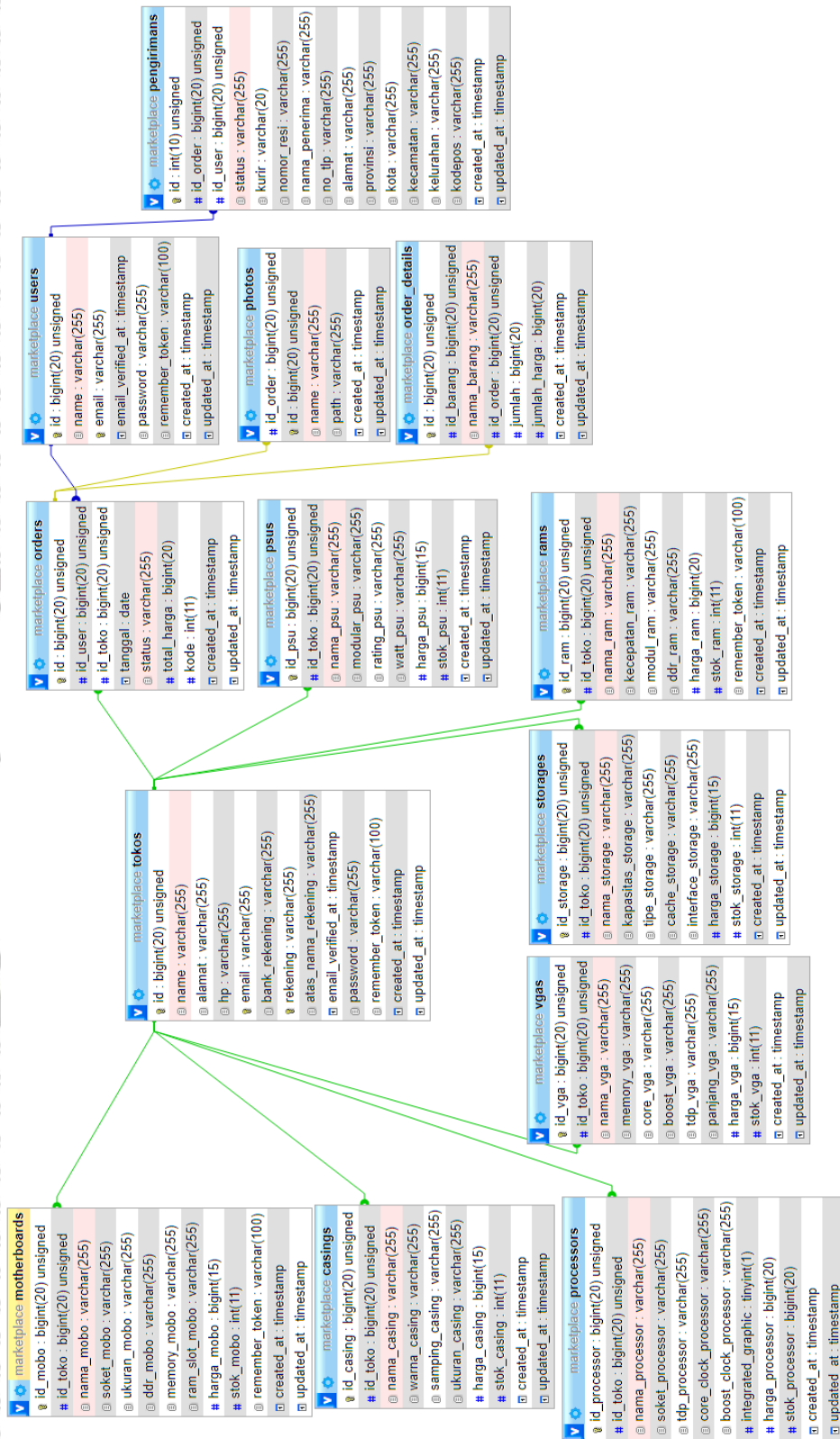
**Gambar 5.34 Implementasi antarmuka isi pengiriman**

Gambar 5.34 merupakan hasil aplikasi antarmuka isi pengiriman untuk toko komputer. Pada halaman ini terdapat *breadcrumb* yang menjadi petunjuk kepada toko mengenai keberadaannya saat itu. Selain itu terdapat sebuah formulir data kurir pengiriman yang berisi nama kurir dan nomor resi yang juga disertai dengan tombol “Simpan” untuk sistem menyimpan data pengiriman dan menampilkan pada halaman detail transaksi pada toko dan detail pesanan pada pembeli.

#### 5.2.2 Implementasi Database

Pada tahap ini akan menghasilkan gambar relasi antar tabel pada database yang sudah diimplementasikan dalam bentuk diagram physical data model. Seperti pada Gambar 5.35.





Gambar 5.35 Implementasi database

### 5.2.3 Implementasi Kode

Implementasi kode merupakan hasil penerapan dari hasil rancangan komponen pada tahap perancangan sistem. Pada sistem penjualan dengan rekomendasi kompatibilitas komponen toko komputer terdapat 3 pokok *method* yang dirancang pada perancangan komponen, yaitu *method* create(), *method* simulasi(), dan *method* bayar().

#### 5.2.3.1 Implementasi Kode pada *Method* create()

Implementasi kode pada *method* create() dibuat pada controller ProcessorController, adapun hasil dari implementasi kode pada *method* create() disajikan pada tabel 5.18.

**Tabel 5.18 Implementasi *method* create()**

No	<i>Method</i> create()
1	function create(Request \$request){
2	//validate input
3	\$request->validate([
4	'nama_processor'=>'required min:5 max:50',
5	'socket_processor'=>'required',
6	'tdp_processor' =>'required',
7	'core_clock_processor' =>'required',
8	'boost_clock_processor' =>'required',
9	'integrated_graphic'=>'required boolean',
10	'harga_processor'=>'required',
11	'stok_processor'=>'required',
12	]);
13	
14	\$proc = new Processor();
15	\$proc->id_toko = \$request->user()->id;
16	\$proc->id_processor = mt_rand(1000, 9999);
17	\$proc->nama_processor = \$request->
18	nama_processor;
19	\$proc->socket_processor = \$request->
20	socket_processor;
21	\$proc->tdp_processor = \$request->
22	tdp_processor;
	\$proc->core_clock_processor = \$request->
	core_clock_processor;



```

23 $proc->boost_clock_processor = $request-
    >boost_clock_processor;
24
25 $proc->integrated_graphic = $request-
    >integrated_graphic;
26
27 $proc->harga_processor = $request-
    >harga_processor;
28
29 $proc->stok_processor = $request-
    >stok_processor;
30
31 $save = $proc->save();
32 if($save) {
33     Returnredirect()->back()-
    >with('success','Barang Telah berhasil
    Dimasukkan');
34 }
35 }
36 }
37

```

### 5.2.3.2 Implementasi Kode pada *Method* simulasi()

Implementasi kode pada *method* simulasi() dibuat pada controller SimulasiController, adapun hasil dari implementasi kode pada *method* create() disajikan pada tabel 5.19.

**Tabel 5.19 Implementasi *method* simulasi()**

No	<i>Method</i> simulasi()
1	Public function simulasi() {
2	
3	\$total_harga =
4	Session::get('harga_processor')+
5	Session::get('harga_mobo)+
6	Session::get('harga_ram')+
7	Session::get('harga_vga')+
8	Session::get('harga_storage')+
9	Session::get('harga_psu')+
10	Session::get('harga_casing');
11	
12	
13	\$total_hara = number_format(\$total_harga);
14	
15	\$tdp_komputer = Session::get('tdp_processor')
16	+ Session::get('tdp_vga')+ 150;
17	

18	Session::put('tdp_komputer', \$tdp_komputer);
19	Session::put('total_harga', \$total_harga);
20	return view
21	('dashboard.user.simulasi.simulasi');
22	}

### 5.2.3.3 Implementasi Kode pada *Method* bayar()

Implementasi kode pada *method* create() dibuat pada controller ProcessorController, adapun hasil dari implementasi kode pada *method* create() disajikan pada tabel 5.20.

**Tabel 5.20 Implementasi *method* bayar()**

No	<i>Method</i> create()
1	public function bayar (\$id)
2	{
3	\$photo = Photo::where('id_order', \$id)-
4	>first();
5	
6	if(empty(\$photo)){
7	return redirect()->back()->with('fail',
8	'Foto Bukti Transfer belum diupload, Mohon
9	untuk upload terlebih dahulu');
10	}
11	else {
12	order = Order::where('id', \$id)->first();
13	\$order->status = "menunggu Konfirmasi
14	Pembayaran";
15	\$order->update();
16	}
17	Return \$this->index();
18	}



## BAB 6 PENGUJIAN SISTEM

Pada bagian ini, penulis akan membahas tahapan untuk menguji sistem dengan meninjau apakah sistem yang telah dirancang dan dibuat dapat berfungsi sesuai dengan kebutuhan yang telah didefinisikan. Terdapat 2 jenis teknik pengujian pada tahap ini, yaitu dengan *white box testing* dan *black box testing*.

### 6.1 Pengujian Unit

Pada pengujian unit, penulis menguji 3 *method* yang ada pada Sistem Penjualan dengan Rekomendasi Kompatibilitas Komponen pada Toko Komputer, yaitu *method create()* yang dijalankan pada saat toko menekan tombol “tambah barang” pada formulir tambah produk *processor*, *method simulasi()* yang dijalankan pada saat *member* memilih komponen komputer pada halaman simulasi, dan *method bayar()* yang dijalankan pada saat *member* sudah melakukan pembayaran dan ingin menekan tombol “sudah transfer” pada halaman detail transaksi.

#### 6.1.1 Pengujian Unit *Class ProcessorController Method Create()*

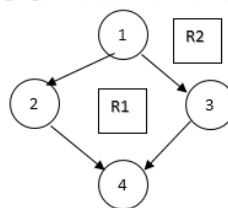
##### 1. Pseudocode

Tabel 6.1 Pseudocode *method create()*

No	Pseudocode
1	Start
1	input data processor
1	validate input
1	if (validate) {
2	Create new processor
2	Data processor = input
3	} else {
3	Return error message invalid input
3	End if
4	End

##### 2. Basic Path Testing

###### a) Flow Graph



Gambar 6.1 Grafik *flow method create()*

###### b) Cyclomatic Complexity

$$V(G) = \text{Jumlah Region} = 2$$

$$V(G) = \text{Edge} - \text{Node} + 2 = 4 - 4 + 2 = 2$$

$$V(G) = \text{Predicate} + 1 = 1 + 1 = 2$$

c) Independent Path

1. Jalur 1: 1-2-4

2. Jalur 2: 1-3-4

d) Pengujian Unit Menambah Barang Jualan (*Processor*)

**Tabel 6.2 Kasus uji *method* menambah barang jualan**

No	Nomor jalur	Masukkan data	Hasil Harapan	Hasil	Status
1	1	<ul style="list-style-type: none"> <li>- Pertama method driver2() mengeset nilai parameter nama_processor dengan i5 6400.</li> <li>- Kedua method driver2() mengeset nilai parameter soket_processor dengan 1156.</li> <li>- Ketiga method driver2() mengeset nilai parameter tdp_processor dengan 65</li> <li>- Keempat method driver2() mengeset nilai parameter core_clock_processor dengan 3.0</li> <li>- Kelima method driver2() mengeset nilai parameter boost_clock_processor dengan 3.6.</li> <li>- Keenam method driver2() mengeset nilai parameter integrated_graphic dengan nilai 1</li> </ul>	Data barang telah berhasil dimasukkan	Data barang telah berhasil dimasukkan	Valid
2	2	<ul style="list-style-type: none"> <li>- Pertama method driver2() mengeset nilai parameter</li> </ul>	Data tidak tersimpan	Data tidak tersimpan	Valid



		nama_processor dengan i5 6400.		
		- Kedua method driver2() mengeset nilai parameter socket_processor dengan 1156.		
		- Ketika method driver2() mengeset nilai parameter tdp_processor dengan 65		
		- Keempat method driver2() mengeset nilai parameter core_clock_processor dengan 3.0		
		- Kelima method driver2() mengeset nilai parameter boost_clock_processo r dengan 3.6		
		- Keenam method driver2() mengeset nilai parameter integrated_graphic dengan 12341.		

Untuk *method* driver2() pada pengujian method create() disajikan pada Gambar 6.2.

```

public function driver2(Request $request)
{
    $request->request->set('nama_processor', 'I5 6400');
    $request->request->set('soket_processor', '1156');
    $request->request->set('tdp_processor', '65');
    $request->request->set('core_clock_processor', '3.0');
    $request->request->set('boost_clock_processor', '3.6');
    //kondisi benar
    // $request->request->set('integrated_graphic', '1');
    //kondisi salah
    $request->request->set('integrated_graphic', '12341');
    $request->request->set('harga_processor', '1500000');
    $request->request->set('stok_processor', '10');
    //validate input
    $request->validate([
        'nama_processor' => 'required|min:5|max:50',
        'soket_processor' => 'required',
        'tdp_processor' => 'required',
        'core_clock_processor' => 'required',
        'boost_clock_processor' => 'required',
        'integrated_graphic' => 'required|boolean',
        // 'gambar_processor' => 'required|image|mimes:jpeg,png,jpg|max:5000',
        'harga_processor' => 'required',
        'stok_processor' => 'required',
    ]);

    $proc = new Processor();
    $proc->id_toko = $request->user()->id;
    $proc->id_processor = mt_rand(1000,9999);
    $proc->nama_processor = $request->nama_processor;
    $proc->soket_processor = $request->soket_processor;
    $proc->tdp_processor = $request->tdp_processor;
    $proc->core_clock_processor = $request->core_clock_processor;
    $proc->boost_clock_processor = $request->boost_clock_processor;
    $proc->integrated_graphic = $request->integrated_graphic;
    // $proc->gambar_processor = $request->gambar_processor;
    $proc->harga_processor = $request->harga_processor;
    $proc->stok_processor = $request->stok_processor;
    $save = $proc->save();

    if($save){
        return redirect()->back()->with('success','Barang Telah berhasil Dimasukkan');
    } else
        echo $request;
}

```

Gambar 6.2 Method driver2() untuk uji unit method create()

## 6.1.2 Pengujian Unit Class SimulasiController Method Simulasi

### 1. Pseudocode

Tabel 6.3 Pseudocode method simulasi()

No	Pseudocode
1	Start
1	Session data komponen
1	\$total_harga = session data (harga_processor +
1	harga_motherboard + harga_ram + harga_vga +
1	harga_stirage + harga_psu + harga_casing)
1	
1	\$tdp_komputer = session data (tdp_processor +
1	tdp_vga) + 150
1	
1	End



## 2. Basic Path Testing

### a) Flow Graph



**Gambar 6.3 Grafik flow method simulasi()**

### b) Cyclomatic Complexity

$V(G) = \text{Jumlah Region} = 1$

$V(G) = \text{Edge} - \text{Node} + 2 = 1 - 2 + 2 = 1$

$V(G) = \text{Predicate} + 1 = 0 + 1 = 1$

### c) Independent Path

1. Jalur 1: 1 – 2

### d) Pengujian Mengisi Halaman Simulasi

**Tabel 6.4 Kasus uji mengisi halaman simulasi**

No	Nomor jalur	Masukkan data	Hasil Harapan	Hasil	Status
1	1	<ul style="list-style-type: none"> <li>- Pertama kelas driver memanggil method <code>saveProcessor()</code> dengan parameter ("1", "1", "I7 6700", "1156", "4300000", "65").</li> <li>- Kedua kelas driver memanggil method <code>saveMobo()</code> dengan parameter ("1", "1", "Asus ROG rampage VI apex", "1156", "7850000").</li> <li>- Ketiga kelas driver memanggil method <code>saveRam()</code> dengan parameter ("1", "1", "Klevv DDR4", "1400000").</li> <li>- Keempat kelas driver memanggil method</li> </ul>	Processor : Intel Core I7 6700  Harga : 4300000  Motherboa rd : Asus ROG rampage VI apex  Harga : 7850000  RAM : Klevv DDr4 Bolt Serries PC19200  Harga : 1400000	Processor : Intel Core I7 6700  Harga : 4300000  Motherboa rd : Asus ROG rampage VI apex  Harga : 7850000  RAM : Klevv DDR4 Bolt Serries PC19200  Harga : 1400000	Valid

	saveVga() dengan parameter ("1", "1", "Zotac RTX3060", "13750000", "170").	Vga Card : Zotac RTX 3060	Vga Card : Zotac RTX 3060
	- Kelima kelas driver memanggil method saveStorage() dengan parameter ("1", "1", "SSD Patriot Brust", "1250000").	Harga : 13750000	Harga : 13750000
	- Keenam kelas driver memanggil method savePsu() dengan parameter ("1", "1", "Xigmatek Minotaur 650 W", "1300000", "650").	Storage : SSD Patriot Brust 2.5 Inch sata 3.0	Storage : SSD Patriot Brust 2.5 Inch sata 3.0
	- Ketujuh kelas driver memanggil method saveCasing() dengan parameter ("1", "1", "Cooler Master Masterbox MB530P", "1650000").	Harga : 1250000	Harga : 1250000
	1. Terakhir kelas driver memanggil method simulasi().	Casing : Cooler Master Masterbox MB530P	Casing : Cooler Master Masterbox MB530P
		Harga : 1650000	Harga : 1650000
		Power Supply : Xigmatek Minotaur 650W	Power Supply : Xigmatek Minotaur 650W
		Harga : 1300000	Harga : 1300000
		Total Prediksi Watt Komputer : 385 Watt	Total Prediksi Watt Komputer : 385 Watt
		Total Harga : 31500000	Total Harga : 31500000

Untuk *method* driver() pada pengujian *method* simulasi() disajikan pada Gambar 6.4 dibawah ini.



```
public function driver()
{
    $processor = (new SimulasiController)
        ->saveProcessor("1", "1", "I7 6700", "1156", "4300000", "65");
    $motherboard = (new SimulasiController)
        ->saveMobo("1", "1", "Asus ROG rampage VI apex", "1156", "7850000");
    $ram = (new SimulasiController)
        ->saveRam("1", "1", "Klevv DDR4", "1400000");
    $vga = (new SimulasiController)
        ->saveVga("1", "1", "Zotac RTX3060", "13750000", "170");
    $storage = (new SimulasiController)
        ->saveStorage("1", "1", "SSD Patriot Brust", "1250000");
    $psu = (new SimulasiController)
        ->savePsu("1", "1", "Xigmatek Minotaur 650 W", "1300000", "650");
    $casing = (new SimulasiController)
        ->saveCasing("1", "1", "Cooler Master Masterbox MB530P", "1650000");

    $simulasi = (new SimulasiController)->simulasi();
}
```

Gambar 6.4 Method driver() untuk uji unit method simulasi()

### 6.1.3 Pengujian Unit Class History Controller Method Bayar

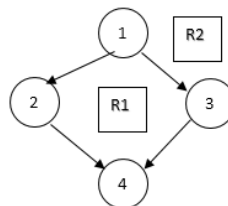
#### 1. Pseudocode

Tabel 6.5 Pseudocode method bayar()

No	Pseudocode
1	Start
1	\$id_order
1	Photo = photo where (id_foto = id_order)
1	if (no photo found)
1	"Foto Bukti Transfer Belum diupload, Mohon
1	untuk Upload terlebih dahulu"
1	Else
1	Update status order = "Menunggu Konfirmasi
1	Pembayaran"
1	endif
1	End

#### 2. Basic Path Testing

##### a) Flow Graph



Gambar 6.5 Grafik flow method bayar()

##### b) Cyclomatic Complexity

$$V(G) = \text{Jumlah Region} = 2$$

$$V(G) = \text{Edge} - \text{Node} + 2 = 4 - 4 + 2 = 2$$

$$V(G) = \text{Predicate} + 1 = 1 + 1 = 2$$

c) *Independent Path*

1. Jalur 1: 1 – 2 – 4

2. Jalur 2: 1 – 3 – 4

d) *Pengujian Unit Bayar*

**Tabel 6.6 Kasus uji unit bayar**

No	Nomor jalur	Masukkan data	Hasil Harapan	Hasil	Status
1	1	<i>Method driver1</i> memanggil method bayar dengan parameter \$id yang bernilai 90.	Status transaksi berubah menjadi “Menunggu Konfirmasi Pembayaran”	Status Transaksi berubah menjadi “Menunggu konfirmasi pembayaran”	Valid
2	2	<i>Method driver1</i> memanggil method bayar dengan parameter \$id yang bernilai 92.	Menampilkan pesan “Foto bukti transfer belum diupload, mohon untuk upload terlebih dahulu.”	Menampilkan pesan “Foto bukti transfer belum diupload, mohon untuk upload terlebih dahulu.”	Valid

Untuk *method driver1()* pada pengujian *method bayar()*, disajikan pada Gambar 6.6.

```
public function driver1()
{
    //kondisi foto ditemukan
    //$id = "90";
    //kondisi foto tidak ditemukan
    $id = "92";

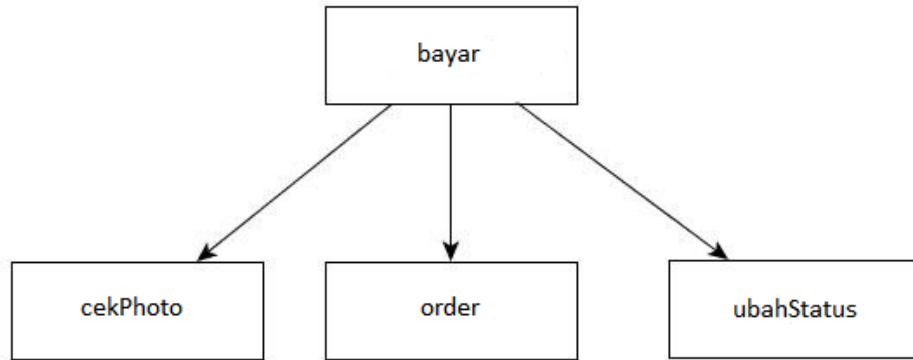
    $bayar = (new HistoryController)->bayar($id);
    $halaman = (new HistoryController)->detail($id);
    echo $halaman;
}
```

**Gambar 6.6 Method driver1() untuk uji unit method bayar()**



## 6.2 Pengujian Integrasi

Pada tahapan pengujian integrasi, dilakukan pengujian pada hubungan satu kelas dengan kelas lainnya. Adapun method yang diujikan adalah bayar(). Pada pengujian ini, digunakan *basis path testing* dan metode *top down*. Adapun diagram hierarki pengujian integrasi tersajikan pada Gambar 6.7.



Gambar 6.7 Diagram hierarki dari pengujian integrasi *method* bayar(\$id)

Tabel 6.7 Identifikasi *method* bayar(\$id)

No	Langkah Uji	Keterangan
1	bayar()+cekPhoto()+order+ubahStatus()	Method bayar digunakan untuk mengubah nilai status order dari "Menunggu Pembayaran" menjadi "Menunggu Konfirmasi Pembayaran" yang dilakukan oleh member. Method bayar() yang berlokasi di kelas HistoryController dijalankan untuk melakukan uji method dari stub_cekPhoto dan model stub_Order. Pengujian dilakukan dengan cara memberikan nilai yang bertipe data string pada stub_cekPhoto untuk menunjukkan kondisi bila photo ditemukan dan kondisi bila photo pada order tidak ditemukan.

### 1. Pseudocode

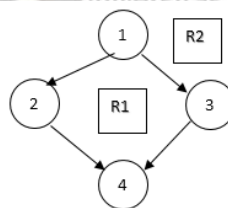
Tabel 6.8 Pseudocode bayar(\$id)

No	Pseudocode
1	Start
2	\$id_order
3	Photo = photo where (id_foto = id_order) If (no photo found)

4	"Foto Bukti Transfer Belum diupload, Mohon
5	untuk Upload terlebih dahulu"
6	Else
7	Update status order = "Menunggu Konfirmasi
8	Pembayaran"
9	endif
10	endif

## 2. Basic Path Testing

### a) Flow Graph method bayar



Gambar 6.8 Grafik *flow method* bayar()

### b) Cyclomatic Complexity method bayar

$V(G) = \text{Jumlah Region} = 2$

$V(G) = \text{Edge} - \text{Node} + 2 = 4 - 4 + 2 = 2$

$V(G) = \text{Predicate} + 1 = 1 + 1 = 2$

### c) Independent Path bayar

1. Jalur 1: 1 – 2 – 4

2. Jalur 2: 1 – 3 – 4

### d) Pengujian Unit Integrasi Bayar

Tabel 6.9 Kasus uji unit *method* bayar(\$id)

No	Nomor jalur	Masukkan data	Hasil Harapan	Hasil	Status
1	1	Method driver1 memanggil method bayar dengan parameter \$id yang bernilai 90.	Status transaksi berubah menjadi "Menunggu Konfirmasi Pembayaran"	Status Transaksi berubah menjadi "Menunggu konfirmasi pembayaran"	Valid
2	2	Method driver1 memanggil method bayar dengan	Menampilkan pesan "Foto bukti transfer belum diupload,	Menampilkan pesan "Foto bukti transfer belum	Valid



	parameter \$id yang bernilai 92.	mohon untuk upload terlebih dahulu.”	diupload, mohon untuk upload terlebih dahulu.”	
--	----------------------------------	--------------------------------------	--	--

Langkah pertama pengujian integrasi dengan menggunakan metode *top down* dengan menguji *method* bayar, cekPhoto dan ubahStatus pada kelas *controller* TestingController. Pada method cekPhoto, akan diberikan 2 nilai berupa nilai null dan 1 yang menggambarkan tidak ditemukan foto bukti transfer yang telah diunggah oleh *member*. Adapun *source code* kondisi benar tersajikan pada Tabel 6.10.

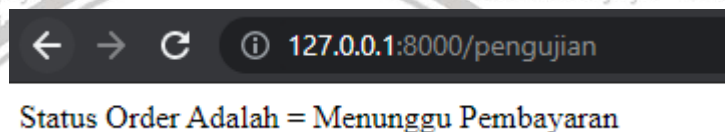
**Tabel 6.10 Source code kondisi foto tidak ditemukan**

```

class TestingController extends Controller
{
    private function stub_cekPhoto()
    {
        return null;
    }
    private function stub_order($status)
    {
        echo "Status Order Adalah = ";
        echo $status;
    }
    private function stub_ubahStatus($status)
    {
        $orderStatus = $status;
        return $this->stub_order($orderStatus);
    }
    public function bayar()
    {
        $photo = $this->stub_cekPhoto();
        if(empty($photo)){
            $status = "Menunggu Pembayaran";
        }
        else {
            $status = "Menunggu Konfirmasi
Pembayaran";
        }
        return $this->stub_ubahStatus($status);
    }
}

```

Hasil dari pengujian pertama dengan memberikan nilai pada *method* *stub\_cekPhoto* dengan nilai null maka pada *method* *bayar* nilai null di set sebagai nilai dari variabel *\$photo*. Kemudian masuk ke konfisi if dimana bila nilai dari variabel *\$photo* adalah kosong/*null* maka akan memberikan nilai “Menunggu Pembayaran” kepada variabel *\$status* yang akan meneruskan nilai dari *\$status* ke *method* *stub\_ubahStatus* yang kemudian memberikan nilai tersebut kepada variabel *\$orderStatus*. Yang kemudian meneruskan variabel tersebut ke kelas model *Order* dan menampilkan pesan “Status Order Adalah = ” beserta nilai dari *\$status*. Hasil dari pengujian pertama tersajikan pada Gambar 6.9.



**Gambar 6.9 Interpretasi hasil pengujian integrasi pertama**

Langkah berikutnya pengujian integrasi dilakukan menggunakan metode *top down* dengan menguji *method* *bayar*, *cekPhoto* dan *ubahStatus* pada kelas *controller* *TestingController*. Pada *method* *cekPhoto* akan diberikan 2 nilai yaitu nilai null dan 1 yang menggambarkan ditemukan foto bukti transfer yang telah diunggah oleh *member*. Adapun *source code* kondisi benar tersajikan pada Tabel 6.11.

**Tabel 6.11 Source code kondisi foto ditemukan**

```
class TestingController extends Controller
{
    private function stub_cekPhoto()
    {
        return 1;
    }
    private function stub_order($status)
    {
        echo "Status Order Adalah = ";
        echo $status;
    }
    private function stub_ubahStatus($status)
    {
        $orderStatus = $status;
        return $this->stub_order($orderStatus);
    }
}
```

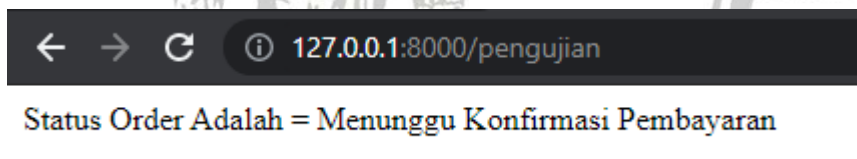


```

public function bayar()
{
    $photo = $this->stub_cekPhoto();
    if(empty($photo)){
        $status = "Menunggu Pembayaran";
    }
    else {
        $status = "Menunggu Konfirmasi
Pembayaran";
    }
    return $this->stub_ubahStatus($status);
}

```

Hasil dari pengujian pertama dengan memberikan nilai pada *method* *method* *stub\_cekPhoto* dengan nilai 1 maka pada *method* *bayar* nilai 1 di set sebagai nilai dari variabel *\$photo* . Kemudian masuk ke konfisi if dimana bila nilai dari variabel *\$photo* adalah kosong/*null* dikarenakan pada pengujian kedua nilai photo diberikan 1 dan tidak *null* maka akan memberikan nilai “Menunggu Konfirmasi Pembayaran” kepada variabel *\$status* yang akan meneruskan nilai dari *\$status* ke *method* *stub\_ubahStatus* yang kemudian memberikan nilai tersebut kepada variabel *\$orderStatus*. Yang kemudian meneruskan variabel tersebut ke kelas model *Order* dan menampilkan pesan “Status Order Adalah = “ beserta nilai dari *\$status*. Interpretasi hasil pengujian pertama tersajikan pada Gambar 6.10.



Gambar 6.10 Integrasi hasil pengujian integrasi kedua

### 6.3 Pengujian Validasi

Pengujian validasi dilakukan untuk menentukan kesesuaian kebutuhan fungsional dengan tujuan yang diharapkan. Pada pengujian validasi, dilakukan pengujian 25 kebutuhan fungsional yang terdapat pada sistem penjualan dengan rekomendasi kompatibilitas komponen pada Toko Komputer.

#### 6.3.1 Pengujian Validasi User Login

Tabel 6.12 Kasus uji validasi login

Kode Kebutuhan	STMK_0100
Nama Kasus Uji	Login berhasil

<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. <i>User</i> menekan tombol “login”.</li> <li>2. Kemudian, <i>user</i> menekan tombol “login” sebagai <i>member</i> atau sebagai toko.</li> <li>3. <i>User</i> memasukkan email dan password pada halaman formulir <i>login</i>.</li> <li>4. <i>User</i> menekan tombol “login”.</li> </ol>
<b>Hasil yang Diharapkan</b>	Aktor berhasil masuk ke dalam sistem sebagai <i>member</i> atau toko komputer.
<b>Hasil Pengujian</b>	Aktor berhasil masuk ke dalam sistem sebagai <i>member</i> atau toko komputer
<b>Status Pengujian</b>	Valid

**Tabel 6.13 Kasus uji validasi alternatif *login 3a***

<b>Kode Kebutuhan</b>	STMK_0100
<b>Nama Kasus Uji</b>	Login pada saat <i>User</i> tidak mengisi salah satu kolom yang ada di formulir.
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. <i>User</i> menekan tombol “Login”</li> <li>2. <i>User</i> menekan tombol “Login” sebagai <i>member</i> atau sebagai toko</li> <li>3. <i>User</i> tidak mengisikan salah satu kolom <i>e-mail</i> atau <i>password</i></li> <li>4. <i>User</i> menekan tombol “Login”</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem menampilkan pesan peringatan “harap isi bidang ini”
<b>Hasil Pengujian</b>	Sistem menampilkan pesan peringatan “harap isi bidang ini”
<b>Status Pengujian</b>	Valid

**Tabel 6.14 Kasus uji validasi alternatif *login 3b***

<b>Kode Kebutuhan</b>	STMK_0100
<b>Nama Kasus Uji</b>	Login pada saat <i>User</i> mengisikan email yang tidak terdaftar pada sistem.
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. <i>User</i> berada pada halaman <i>login</i></li> <li>2. <i>User</i> memasukkan <i>e-mail</i> dan <i>password</i> tidak terdaftar pada sistem</li> </ol>



	3. <i>User</i> menekan tombol “Login”
<b>Hasil yang diharapkan</b>	Sistem akan menampilkan pesan peringatan “E-mail Anda belum terdaftar”
<b>Hasil Pengujian</b>	Sistem akan menampilkan pesan peringatan “E-mail Anda belum terdaftar”
<b>Status Pengujian</b>	Valid

### 6.3.2 Pengujian Validasi *Logout*

Tabel 6.15 Kasus uji validasi *logout*

<b>Kode Kebutuhan</b>	STMK_0200
<b>Nama Kasus Uji</b>	<i>Logout</i> berhasil
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Berada pada halaman utama sistem.</li> <li>2. Menekan tombol “logout”.</li> <li>3. Sistem menampilkan pesan konfirmasi <i>logout</i>.</li> <li>4. Menekan tombol “ya” pada pesan konfirmasi <i>logout</i>.</li> </ol>
<b>Hasil yang diharapkan</b>	Aktor berhasil keluar dari sistem
<b>Hasil Pengujian</b>	Aktor berhasil keluar dari sistem
<b>Status Pengujian</b>	Valid

### 6.3.3 Pengujian Validasi *Register*

Tabel 6.16 Kasus uji validasi register

<b>Kode Kebutuhan</b>	STMK_0300
<b>Nama Kasus Uji</b>	Registrasi berhasil
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. <i>User</i> menekan tombol <i>create new account</i> yang ada di halaman <i>login member</i></li> <li>2. <i>User</i> mengisi formulir dengan memasukkan data nama, <i>e-mail</i>, <i>password</i>, dan konfirmasi <i>password</i></li> <li>3. <i>User</i> menekan tombol “register”</li> </ol>
<b>Hasil yang diharapkan</b>	Data <i>User</i> telah berhasil disimpan oleh sistem.
<b>Hasil Pengujian</b>	Data <i>User</i> telah berhasil disimpan oleh sistem.

Status Pengujian	Valid
------------------	-------

Tabel 6.17 Kasus uji validasi alternatif register 3a

Kode Kebutuhan	STMK_0300
Nama Kasus Uji	Registrasi pada saat User tidak mengisi salah satu kolom yang ada di formulir.
Prosedur	<ol style="list-style-type: none"> <li>1. User menekan tombol <i>create new account</i> yang ada di halaman <i>login member</i></li> <li>2. User tidak mengisikan salah satu kolom pada formulir berupa data nama, <i>e-mail</i>, <i>password</i>, dan konfirmasi <i>password</i></li> <li>3. User menekan <i>button</i> "register"</li> </ol>
Hasil yang diharapkan	Sistem menampilkan pesan peringatan "harap diisi form ini"
Hasil Pengujian	Sistem menampilkan pesan peringatan "harap diisi form ini"
Status Pengujian	Valid

#### 6.3.4 Pengujian Validasi Melihat Halaman Panduan

Tabel 6.18 Kasus uji validasi melihat halaman panduan

Kode Kebutuhan	STMK_0400
Nama Kasus Uji	Melihat halaman panduan.
Prosedur	1. User menekan logo "Marketplace Komputer"
Hasil yang diharapkan	Sistem menampilkan halaman panduan.
Hasil Pengujian	Sistem menampilkan halaman panduan.
Status Pengujian	Valid



### 6.3.5 Pengujian Validasi Melihat Daftar Komponen

Tabel 6.19 Kasus uji validasi melihat daftar komponen

Kode Kebutuhan	STMK_0500
Nama Kasus Uji	Melihat daftar komponen <i>processor</i> .
Prosedur	1. <i>User</i> menekan tombol menu “Kategori Komponen” 2. <i>User</i> menekan tombol <i>processor</i> .
Hasil yang diharapkan	Sistem menampilkan halaman daftar komponen <i>processor</i> .
Hasil Pengujian	Sistem menampilkan halaman daftar komponen <i>processor</i> .
Status Pengujian	Valid

### 6.3.6 Pengujian Validasi Melihat Halaman Simulasi

Tabel 6.20 Kasus uji validasi melihat halaman simulasi

Kode Kebutuhan	STMK_0600
Nama Kasus Uji	Melihat halaman simulasi.
Prosedur	1. <i>User</i> menekan tombol menu “Simulasi”
Hasil yang diharapkan	Sistem menampilkan halaman simulasi.
Hasil Pengujian	Sistem menampilkan halaman simulasi.
Status Pengujian	Valid

Tabel 6.21 Kasus uji validasi alternatif melihat halaman simulasi 2a

Kode Kebutuhan	STMK_0600
Nama Kasus Uji	Ketika member belum melakukan login saat melihat halaman simulasi
Prosedur	1. <i>User</i> menekan tombol menu “Simulasi”
Hasil yang diharapkan	Sistem menampilkan halaman <i>login</i> .
Hasil Pengujian	Sistem menampilkan halaman <i>login</i> .
Status Pengujian	Valid

### 6.3.7 Pengujian Validasi Pengisian Halaman Simulasi

Tabel 6.22 Kasus uji validasi mengisi halaman simulasi

Kode Kebutuhan	STMK_0700
Nama Kasus Uji	Mengisi halaman simulasi.





<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol “+” pada baris <i>Processor</i></li> <li>2. Aktor menekan tombol “+” pada baris <i>Processor</i> yang diinginkan</li> <li>3. Aktor menekan tombol “+” pada baris <i>Motherboard</i></li> <li>4. Aktor menekan tombol “+” pada baris <i>Motherboard</i> yang diinginkan</li> <li>5. Aktor menekan tombol “+” pada baris RAM</li> <li>6. Aktor menekan tombol “+” pada baris RAM yang diinginkan</li> <li>7. Aktor menekan tombol “+” pada baris VGA <i>Card</i></li> <li>8. Aktor menekan tombol “+” pada baris VGA <i>Card</i> yang diinginkan</li> <li>9. Aktor menekan tombol “+” pada baris <i>Storage</i></li> <li>10. Aktor menekan tombol “+” pada baris <i>Storage</i> yang diinginkan</li> <li>11. Aktor menekan tombol “+” pada baris <i>Casing</i></li> <li>12. Aktor menekan tombol “+” pada baris <i>Casing</i> yang diinginkan</li> <li>13. Aktor menekan tombol “+” pada baris <i>Power Supply</i></li> <li>14. Aktor menekan tombol “+” pada baris <i>Power Supply</i> yang diinginkan.</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem menampilkan halaman yang berisikan pilihan komponen komputer yang telah dipilih oleh member.

<b>Hasil Pengujian</b>	Sistem menampilkan halaman yang berisikan pilihan komponen komputer yang telah dipilih oleh <i>member</i> .
<b>Status Pengujian</b>	Valid

### 6.3.8 Pengujian Validasi Memasukkan Barang ke Keranjang

Tabel 6.23 Kasus uji validasi memasukkan barang ke keranjang

<b>Kode Kebutuhan</b>	STMK_0800
<b>Nama Kasus Uji</b>	Memasukkan barang ke keranjang.
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. <i>Member</i> berada pada halaman daftar komponen.</li> <li>2. <i>Member</i> memasukkan angka berapa produk yang ingin dibeli.</li> <li>3. <i>Member</i> menekan tombol “+” pada baris produk.</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem berhasil menambahkan barang yang dipilih oleh <i>member</i> ke dalam keranjang.
<b>Hasil Pengujian</b>	Sistem berhasil menambahkan barang yang dipilih oleh <i>member</i> ke dalam keranjang.
<b>Status Pengujian</b>	Valid

Tabel 6.24 Kasus uji validasi alternatif memasukkan barang ke keranjang 1a

<b>Kode Kebutuhan</b>	STMK_0800
<b>Nama Kasus Uji</b>	Memasukkan barang ke keranjang.
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. <i>Member</i> berada pada halaman daftar komponen.</li> <li>2. <i>Member</i> memasukkan angka lebih dari stok barang yang ditampilkan oleh sistem.</li> <li>3. <i>Member</i> menekan tombol “+” pada baris produk.</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem menampilkan pesan peringatan “Pembelian melebihi stok yang ada”.



<b>Hasil Pengujian</b>	Sistem menampilkan pesan peringatan "Pembelian melebihi stok yang ada".
<b>Status Pengujian</b>	Valid

### 6.3.9 Pengujian Validasi Melihat Keranjang

Tabel 6.25 Kasus uji validasi melihat keranjang

<b>Kode Kebutuhan</b>	STMK_0900
<b>Nama Kasus Uji</b>	Melihat keranjang dalam kondisi keranjang terisi
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Aktor berada di halaman daftar produk <i>processor</i>,</li> <li>2. Aktor memasukkan angka barang yang ingin dibeli kemudian menekan tombol "+",</li> <li>3. Aktor menekan menu "Keranjang".</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem memberikan tampilan halaman daftar keranjang dan data nama barang, rincian produk, harga barang, jumlah barang, dan total harga barang.
<b>Hasil Pengujian</b>	Sistem memberikan tampilan halaman daftar keranjang dan data nama barang, rincian produk, harga barang, jumlah barang, dan total harga barang.
<b>Status Pengujian</b>	Valid

Tabel 6.26 Kasus uji validasi alternatif melihat keranjang 2a

<b>Kode Kebutuhan</b>	STMK_0900
<b>Nama Kasus Uji</b>	Melihat keranjang dalam kondisi keranjang tidak terisi
<b>Prosedur</b>	1. Aktor menekan menu "Keranjang".
<b>Hasil yang diharapkan</b>	Sistem memberikan tampilan halaman daftar keranjang dan pesan "Keranjang Anda Kosong"
<b>Hasil Pengujian</b>	Sistem memberikan tampilan halaman daftar keranjang dan pesan "Keranjang Anda Kosong"

Status Pengujian	Valid
------------------	-------

### 6.3.10 Pengujian Validasi Penghapusan Isi Keranjang

Tabel 6.27 Kasus uji validasi penghapusan isi keranjang

Kode Kebutuhan	STMK_1000
Nama Kasus Uji	Menghapus Isi Keranjang.
Prosedur	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol "X" pada kolom yang dipilih pada tabel produk di halaman keranjang,</li> <li>2. Aktor mengkonfirmasi penghapusan barang di keranjang dengan menekan tombol "Ok".</li> </ol>
Hasil yang diharapkan	Data barang yang telah dipilih oleh aktor berhasil dihapus dari keranjang oleh sistem.
Hasil Pengujian	Data barang yang telah dipilih oleh aktor berhasil dihapus dari keranjang oleh sistem.
Status Pengujian	Valid

Tabel 6.28 Kasus uji validasi alternatif penghapusan isi keranjang 4a

Kode Kebutuhan	STMK_1000
Nama Kasus Uji	Menghapus Isi Keranjang.
Prosedur	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol "X" pada kolom yang dipilih pada tabel produk di halaman keranjang,</li> <li>2. Aktor mengkonfirmasi penghapusan barang di keranjang dengan menekan tombol "Cancel".</li> </ol>
Hasil yang diharapkan	Sistem tidak menghapus barang dari keranjang.
Hasil Pengujian	Sistem tidak menghapus barang dari keranjang.
Status Pengujian	Valid



### 6.3.11 Pengujian Mengisi Alamat Pengiriman

Tabel 6.29 Kasus uji validasi mengisi alamat pengiriman

Kode Kebutuhan	STMK_1100
Nama Kasus Uji	Mengisi alamat pengiriman.
Prosedur	<ol style="list-style-type: none"> <li>1. Member menekan tombol “bayar”.</li> <li>2. Member menekan tombol “OK”.</li> <li>3. Member mengisikan <i>form</i> membeli barang yang berisi nama lengkap penerima, nomor telepon penerima, alamat, provinsi, kota, kecamatan, kelurahan, dah kode pos.</li> <li>4. Member menekan tombol “Simpan Alamat Pengiriman”.</li> <li>5. Member menekan tombol “Yes”.</li> </ol>
Hasil yang diharapkan	Data alamat pengiriman berhasil disimpan oleh sistem.
Hasil Pengujian	Data alamat pengiriman berhasil disimpan oleh sistem.
Status Pengujian	Valid

Tabel 6.30 Kasus uji validasi alternatif mengisi alamat pengiriman 3a

Kode Kebutuhan	STMK_1100
Nama Kasus Uji	Membatalkan pengisian alamat pengiriman.
Prosedur	<ol style="list-style-type: none"> <li>1. Member menekan tombol “bayar”.</li> <li>2. Member menekan tombol “Cancel”.</li> </ol>
Hasil yang diharapkan	Sistem membatalkan penampilan <i>form</i> alamat pengiriman.
Hasil Pengujian	Sistem membatalkan penampilan <i>form</i> alamat pengiriman.
Status Pengujian	Valid

Tabel 6.31 Kasus uji validasi alternatif mengisi alamat pengiriman 5a

<b>Kode Kebutuhan</b>	STMK_1100
<b>Nama Kasus Uji</b>	Mengisi alamat pengiriman.
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Member menekan tombol “bayar”.</li> <li>2. Member menekan tombol “OK”</li> <li>3. Member tidak mengisi salah satu <i>form</i> membeli barang yang berisi nama lengkap penerima, nomor telepon penerima, alamat, provinsi, kota, kecamatan, kelurahan, dah kode pos.</li> <li>4. Member menekan tombol “Simpan Alamat Pengiriman”.</li> <li>5. Member menekan tombol “Yes”.</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem memberikan pesan peringatan “harap isi form bagian ini”.
<b>Hasil Pengujian</b>	Sistem menampilkan pesan peringatan “harap isi form bagian ini.
<b>Status Pengujian</b>	Valid

Tabel 6.32 Kasus uji validasi alternatif mengisi alamat pengiriman 8a

<b>Kode Kebutuhan</b>	STMK_1100
<b>Nama Kasus Uji</b>	Mengisi alamat pengiriman.
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Member menekan tombol “bayar”.</li> <li>2. Member menekan tombol “OK”.</li> <li>3. Member mengisi <i>form</i> membeli barang yang berisi nama lengkap penerima, nomor telepon penerima, alamat, provinsi, kota, kecamatan, kelurahan, dah kode pos.</li> <li>4. Member menekan tombol “Simpan Alamat Pengiriman”.</li> <li>5. Member menekan tombol “Cancel”.</li> </ol>



<b>Hasil yang diharapkan</b>	Sistem membatalkan penyimpanan data alamat pengiriman.
<b>Hasil Pengujian</b>	Sistem membatalkan penyimpanan data alamat pengiriman.
<b>Status Pengujian</b>	Valid

### 6.3.12 Pengujian Validasi Pengubahan Alamat Pengiriman

Tabel 6.33 Kasus uji validasi pengubahan alamat pengiriman

<b>Kode Kebutuhan</b>	STMK_1200
<b>Nama Kasus Uji</b>	Mengubah alamat pengiriman.
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. <i>Member</i> menekan tombol “bayar”</li> <li>2. <i>Member</i> menekan tombol “Ok”,</li> <li>3. <i>Member</i> mengisikan form membeli barang yang berisikan nama lengkap penerima, nomor telepon penerima, alamat, provinsi, kota, kecamatan, kelurahan, dan kode pos,</li> <li>4. <i>Member</i> menekan tombol “Simpan Alamat Pengiriman”,</li> <li>5. <i>Member</i> menekan tombol “Yes”.</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem dapat menyimpan data alamat pengiriman yang baru.
<b>Hasil Pengujian</b>	Sistem dapat menyimpan data alamat pengiriman yang baru.
<b>Status Pengujian</b>	Valid

Tabel 6.34 Kasus uji validasi alternatif pengubahan alamat pengiriman 3a

<b>Kode Kebutuhan</b>	STMK_1200
<b>Nama Kasus Uji</b>	Membatalkan penampilan <i>form</i> alamat pengiriman.
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. <i>Member</i> menekan tombol “Ingin mengganti alamat?”</li> </ol>

	2. <i>Member</i> menekan tombol “Cancel”.
<b>Hasil yang diharapkan</b>	Sistem batal menampilkan <i>form</i> alamat pengiriman.
<b>Hasil Pengujian</b>	Sistem batal menampilkan <i>form</i> alamat pengiriman.
<b>Status Pengujian</b>	Valid

**Tabel 6.35 Kasus uji validasi alternatif pengubahan alamat pengiriman 5a**

<b>Kode Kebutuhan</b>	STMK_1200
<b>Nama Kasus Uji</b>	Mengubah alamat pengiriman pada saat <i>member</i> tidak mengisi salah satu kolom pada <i>form</i> .
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. <i>Member</i> menekan tombol “Ingin mengganti alamat?”,</li> <li>2. <i>Member</i> menekan tombol “Ok”,</li> <li>3. <i>Member</i> tidak mengisi salah satu <i>form</i> membeli barang yang berisi nama lengkap penerima, nomor telepon penerima, alamat, provinsi, kota, kecamatan, kelurahan, dan kode pos,</li> <li>4. <i>Member</i> menekan tombol “Simpan Alamat Pengiriman”,</li> <li>5. <i>Member</i> menekan tombol “Yes”.</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem memberikan respon tampilan pesan peringatan “harap isi form bagian ini”.
<b>Hasil Pengujian</b>	Sistem memberikan respon tampilan pesan peringatan “harap isi form bagian ini”.
<b>Status Pengujian</b>	Valid

**Tabel 6.36 Kasus uji validasi alternatif pengubahan alamat pengiriman 8a**

<b>Kode Kebutuhan</b>	STMK_1200
<b>Nama Kasus Uji</b>	Pembatalan pengubahan alamat pengiriman
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. <i>Member</i> menekan tombol “bayar”,</li> </ol>



	<ol style="list-style-type: none"> <li>2. <i>Member</i> menekan tombol “OK”,</li> <li>3. <i>Member</i> mengisikan form membeli barang yang berisikan nama lengkap penerima, nomor telepon penerima, alamat, provinsi, kota, kecamatan, kelurahan, dan kode pos.</li> <li>4. <i>Member</i> menekan tombol “Simpan Alamat Pengiriman”,</li> <li>5. <i>Member</i> menekan tombol “Cancel”.</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem membatalkan penyimpanan data alamat pengiriman.
<b>Hasil Pengujian</b>	Sistem membatalkan penyimpanan data alamat pengiriman.
<b>Status Pengujian</b>	Valid

### 6.3.13 Pengujian Validasi Pembelian Barang

Tabel 6.37 Kasus uji validasi pembelian barang

<b>Kode Kebutuhan</b>	STMK_1300
<b>Nama Kasus Uji</b>	Membeli barang
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. <i>Member</i> menekan tombol “bayar”,</li> <li>2. <i>Member</i> menekan tombol “OK”,</li> <li>3. <i>Member</i> mengisikan <i>form</i> pembelian barang yang berisikan nama lengkap penerima, nomor telepon penerima, alamat, provinsi, kota, kecamatan, kelurahan, dan kode pos,</li> <li>4. <i>Member</i> menekan tombol “Simpan Alamat Pengiriman”,</li> <li>5. <i>Member</i> menekan tombol “Yes”.</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem berhasil mengganti status transaksi dari “Keranjang” menjadi “Menunggu Pembayaran”.

<b>Hasil Pengujian</b>	Sistem berhasil mengganti status transaksi dari “Keranjang” menjadi “Menunggu Pembayaran”.
<b>Status Pengujian</b>	Valid

**Tabel 6.38 Kasus uji validasi alternatif pembelian barang 5a**

<b>Kode Kebutuhan</b>	STMK_1300
<b>Nama Kasus Uji</b>	Membeli barang pada saat member tidak mengisi salah satu kolom pada <i>form</i> .
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. <i>Member</i> menekan tombol “bayar”.</li> <li>2. <i>Member</i> menekan tombol “OK”.</li> <li>3. <i>Member</i> tidak mengisi salah satu <i>form</i> pembelian barang yang berisikan nama lengkap penerima, nomor telepon penerima, alamat, provinsi, kota, kecamatan, kelurahan, dan kode pos.</li> <li>4. <i>Member</i> menekan tombol “Simpan Alamat Pengiriman”.</li> <li>5. <i>Member</i> menekan tombol “Yes”.</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem menampilkan respon pesan peringatan “harap isi form bagian ini”.
<b>Hasil Pengujian</b>	Sistem menampilkan respon pesan peringatan “harap isi form bagian ini”.
<b>Status Pengujian</b>	Valid

**Tabel 6.39 Kasus uji validasi alternatif pembelian barang 8a**

<b>Kode Kebutuhan</b>	STMK_1300
<b>Nama Kasus Uji</b>	Pembatalan Pembelian Barang
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. <i>Member</i> menekan tombol “bayar”.</li> <li>2. <i>Member</i> menekan tombol “OK”.</li> <li>3. <i>Member</i> mengisikan <i>form</i> pembelian barang yang berisikan nama lengkap</li> </ol>



	penerima, nomor telepon penerima, alamat, provinsi, kota, kecamatan, kelurahan, dan kode pos,
	4. <i>Member</i> menekan tombol “Simpan Alamat Pengiriman”,
	5. <i>Member</i> menekan tombol “Cancel”.
<b>Hasil yang diharapkan</b>	Sistem membatalkan pembelian barang.
<b>Hasil Pengujian</b>	Sistem membatalkan pembelian barang.
<b>Status Pengujian</b>	Valid

### 6.3.14 Pengujian Validasi Melihat Daftar Pesanan

**Tabel 6.40 Kasus uji validasi melihat daftar pesanan**

<b>Kode Kebutuhan</b>	STMK_1400
<b>Nama Kasus Uji</b>	Melihat daftar transaksi.
<b>Prosedur</b>	1. <i>Member</i> menekan tombol “Transaksi”
<b>Hasil yang diharapkan</b>	Sistem batal menampilkan <i>form</i> alamat pengiriman.
<b>Hasil Pengujian</b>	Sistem batal menampilkan <i>form</i> alamat pengiriman.
<b>Status Pengujian</b>	Valid

**Tabel 6.41 Kasus uji validasi melihat daftar pesanan 2a**

<b>Kode Kebutuhan</b>	STMK_1400
<b>Nama Kasus Uji</b>	Melihat daftar transaksi dalam kondisi belum login.
<b>Prosedur</b>	1. <i>Member</i> menekan tombol “Transaksi”
<b>Hasil yang diharapkan</b>	Sistem menampilkan halaman login.
<b>Hasil Pengujian</b>	Sistem menampilkan halaman login.
<b>Status Pengujian</b>	Valid

### 6.3.15 Pengujian Validasi Melihat Detail Pesanan

Tabel 6.42 Kasus uji validasi detail pesanan

Kode Kebutuhan	STMK_1500
Nama Kasus Uji	Melihat <i>detail</i> transaksi
Prosedur	1. <i>Member</i> menekan tombol “Transaksi”
Hasil yang diharapkan	Sistem menampilkan halaman <i>detail</i> transaksi.
Hasil Pengujian	Sistem menampilkan halaman <i>detail</i> transaksi.
Status Pengujian	Valid

Tabel 6.43 Kasus uji validasi alternatif melihat detail pesanan 2a

Kode Kebutuhan	STMK_1500
Nama Kasus Uji	Melihat <i>detail</i> transaksi ketika <i>member</i> belum <i>login</i> .
Prosedur	1. <i>Member</i> menekan tombol “Transaksi”
Hasil yang diharapkan	Sistem menampilkan halaman <i>login</i> .
Hasil Pengujian	Sistem menampilkan halaman <i>login</i> .
Status Pengujian	Valid

### 6.3.16 Pengujian Validasi Pembayaran

Tabel 6.44 Kasus uji validasi pembayaran

Kode Kebutuhan	STMK_1600
Nama Kasus Uji	Validasi pembayaran
Prosedur	1. <i>Member</i> menekan tombol “Choose file” dan memasukkan foto bukti <i>transfer</i> . 2. <i>Member</i> menekan tombol “Sudah Transfer”.
Hasil yang diharapkan	Sistem menyimpan foto bukti <i>transfer</i> serta mengubah status pesanan menjadi “Menunggu Konfirmasi Pembayaran”.



<b>Hasil Pengujian</b>	Sistem menyimpan foto bukti <i>transfer</i> serta mengubah status pesanan menjadi “Menunggu Konfirmasi Pembayaran”.
<b>Status Pengujian</b>	Valid

**Tabel 6.45 Kasus uji validasi alternatif pembayaran 3a**

<b>Kode Kebutuhan</b>	STMK_1600
<b>Nama Kasus Uji</b>	Validasi pembayaran tanpa memasukkan foto bukti transfer
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. <i>Member</i> menekan tombol “Choose file”, namun tidak mengisikan foto bukti transfer.</li> <li>2. <i>Member</i> menekan tombol “Sudah Transfer”.</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem menampilkan pesan peringatan “Foto bukti <i>transfer</i> belum di-upload, Mohon untuk upload terlebih dahulu”.
<b>Hasil Pengujian</b>	Sistem menampilkan pesan peringatan “Foto bukti <i>transfer</i> belum di-upload, Mohon untuk upload terlebih dahulu”.
<b>Status Pengujian</b>	Valid

### 6.3.17 Pengujian Validasi Menerima Barang

**Tabel 6.46 Kasus uji validasi menerima barang**

<b>Kode Kebutuhan</b>	STMK_1700
<b>Nama Kasus Uji</b>	Menerima barang
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. <i>Member</i> menekan tombol “Sudah Menerima Barang?”</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem berhasil mengubah status pengiriman menjadi “Sudah Di Terima.”
<b>Hasil Pengujian</b>	Sistem berhasil mengubah status pengiriman menjadi “Sudah Di Terima.”

Status Pengujian	Valid
------------------	-------

**Tabel 6.47 Kasus uji validasi alternatif menerima barang 3a**

Kode Kebutuhan	STMK_1700
Nama Kasus Uji	Pembatalan menerima barang.
Prosedur	1. <i>Member</i> menekan tombol “Sudah Menerima Barang?”.
Hasil yang diharapkan	Sistem membatalkan pengubahan status pengiriman.
Hasil Pengujian	Sistem membatalkan pengubahan status pengiriman.
Status Pengujian	Valid

### 6.3.18 Pengujian Validasi Konfirmasi Pembayaran

**Tabel 6.48 Kasus uji validasi konfirmasi pembayaran**

Kode Kebutuhan	STMK_1800
Nama Kasus Uji	Konfirmasi Pembayaran
Prosedur	1. Toko Komputer menekan tombol “Detail” pada salah satu pesanan yang akan dilakukan konfirmasi pembayaran. 2. Toko Komputer menekan tombol “Sudah Transfer”.
Hasil yang diharapkan	Status pesanan dari “Menunggu Konfirmasi Pembayaran” berhasil diubah oleh sistem menjadi “Lunas”.
Hasil Pengujian	Status pesanan dari “Menunggu Konfirmasi Pembayaran” berhasil diubah oleh sistem menjadi “Lunas”.
Status Pengujian	Valid

**Tabel 6.49 Kasus uji validasi alternatif konfirmasi pembayaran 3a**

Kode Kebutuhan	STMK_1800
----------------	-----------



<b>Nama Kasus Uji</b>	Ketika status pesanan “Lunas” saat melakukan konfirmasi pembayaran
<b>Prosedur</b>	1. Toko Komputer menekan tombol “Detail” pada salah satu pesanan yang akan dilakukan konfirmasi pembayaran.
<b>Hasil yang diharapkan</b>	Tombol “Sudah Transfer” tidak akan ditampilkan oleh sistem.
<b>Hasil Pengujian</b>	Tombol “Sudah Transfer” tidak akan ditampilkan oleh sistem.
<b>Status Pengujian</b>	Valid

### 6.3.19 Pengujian Validasi Melihat Barang Jualan

**Tabel 6.50 Kasus uji validasi melihat barang jualan**

<b>Kode Kebutuhan</b>	STMK_1900
<b>Nama Kasus Uji</b>	Melihat barang jualan
<b>Prosedur</b>	1. Toko Komputer menekan tombol “Your Product”. 2. Toko Komputer menekan tombol pada salah satu kategori yang ingin dilihat.
<b>Hasil yang diharapkan</b>	Halaman data barang jualan sebuah komponen berhasil ditampilkan oleh sistem.
<b>Hasil Pengujian</b>	Halaman data barang jualan sebuah komponen berhasil ditampilkan oleh sistem.
<b>Status Pengujian</b>	Valid

**Tabel 6.51 Kasus uji validasi alternatif melihat barang jualan 3a**

<b>Kode Kebutuhan</b>	STMK_1900
<b>Nama Kasus Uji</b>	Ketika Toko Komputer belum melakukan login saat ingin melihat barang jualan
<b>Prosedur</b>	1. Toko Komputer menekan tombol “Your Product”. 2. Toko Komputer menekan tombol pada salah satu kategori yang ingin dilihat.
<b>Hasil yang diharapkan</b>	Halaman <i>login</i> akan ditampilkan oleh sistem.

<b>Hasil Pengujian</b>	Halaman <i>login</i> akan ditampilkan oleh sistem.
<b>Status Pengujian</b>	Valid

### 6.3.20 Pengujian Validasi Menambah Barang Jualan

Tabel 6.52 Kasus uji validasi menambah barang jualan

<b>Kode Kebutuhan</b>	STMK_2000
<b>Nama Kasus Uji</b>	Mengubah barang jualan
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Toko Komputer menekan tombol “tambah”.</li> <li>2. Toko Komputer mengisi formulir tambah barang jualan yang berisi nama barang, detail barang dan stok barang.</li> <li>3. Toko Komputer menekan tombol “Simpan”.</li> </ol>
<b>Hasil yang diharapkan</b>	Data barang jualan berhasil disimpan oleh sistem.
<b>Hasil Pengujian</b>	Data barang jualan berhasil disimpan oleh sistem.
<b>Status Pengujian</b>	Valid

Tabel 6.53 Kasus uji validasi alternatif menambah barang jualan 1a

<b>Kode Kebutuhan</b>	STMK_2000
<b>Nama Kasus Uji</b>	Ketika salah satu kolom pada formulir tambah barang jualan tidak diisi oleh Toko Komputer
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Toko Komputer menekan tombol “tambah”.</li> <li>2. Toko Komputer tidak mengisi salah satu kolom pada form tambah barang jualan.</li> <li>3. Toko Komputer menekan tombol “Simpan”.</li> </ol>
<b>Hasil yang diharapkan</b>	Pesan peringatan yang berisi “harap isi bidang ini” akan ditampilkan oleh sistem.
<b>Hasil Pengujian</b>	Pesan peringatan yang berisi “harap isi bidang ini” akan ditampilkan oleh sistem.
<b>Status Pengujian</b>	Valid



### 6.3.21 Pengujian Validasi Mengubah Barang Jualan

Tabel 6.54 Kasus uji validasi mengubah barang jualan

<b>Kode Kebutuhan</b>	STMK_2100
<b>Nama Kasus Uji</b>	Mengubah barang jualan
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Toko Komputer menekan tombol “edit” pada baris tabel daftar barang jualan yang dipilih.</li> <li>2. Toko Komputer memasukkan data terbaru pada formulir perubahan data barang jualan.</li> <li>3. Toko Komputer menekan tombol “Update data”.</li> </ol>
<b>Hasil yang diharapkan</b>	Data barang jualan berhasil diperbaharui oleh sistem.
<b>Hasil Pengujian</b>	Data barang jualan berhasil diperbaharui oleh sistem.
<b>Status Pengujian</b>	Valid

Tabel 6.55 Kasus uji validasi alternatif mengubah barang jualan 3a

<b>Kode Kebutuhan</b>	STMK_2100
<b>Nama Kasus Uji</b>	Ketika salah satu kolom pada formulir perubahan barang jualan tidak diisi oleh Toko Komputer
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Toko Komputer menekan tombol “edit” pada baris tabel daftar barang jualan yang dipilih.</li> <li>2. Toko Komputer tidak mengisi salah satu kolom pada <i>form</i> perubahan data barang jualan.</li> <li>3. Toko Komputer menekan tombol “Update Data”.</li> </ol>
<b>Hasil yang diharapkan</b>	Pesan peringatan yang berisi “harap isi bidang ini” akan ditampilkan oleh sistem.
<b>Hasil Pengujian</b>	Pesan peringatan yang berisi “harap isi bidang ini” akan ditampilkan oleh sistem.
<b>Status Pengujian</b>	Valid

### 6.3.22 Pengujian Validasi Menghapus Barang Jualan

**Tabel 6.56 Kasus uji validasi menghapus barang jualan**

<b>Kode Kebutuhan</b>	STMK_2200
<b>Nama Kasus Uji</b>	Menghapus barang jualan
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Toko Komputer menekan tombol “Hapus” pada baris tabel daftar barang jualan yang dipilih.</li> <li>2. Toko Komputer menekan tombol “Yes” saat pesan konfirmasi penghapusan barang jualan ditampilkan oleh sistem.</li> </ol>
<b>Hasil yang diharapkan</b>	Data barang jualan berhasil dihapus oleh sistem.
<b>Hasil Pengujian</b>	Data barang jualan berhasil dihapus oleh sistem.
<b>Status Pengujian</b>	Valid

**Tabel 6.57 Kasus uji validasi alternatif menghapus barang jualan 3a**

<b>Kode Kebutuhan</b>	STMK_2200
<b>Nama Kasus Uji</b>	Membatalkan penghapusan barang jualan
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Toko Komputer menekan tombol “Hapus” pada baris tabel daftar barang jualan yang dipilih.</li> <li>2. Toko Komputer menekan tombol “No” saat pesan konfirmasi penghapusan barang jualan ditampilkan oleh sistem.</li> </ol>
<b>Hasil yang diharapkan</b>	Penghapusan barang jualan berhasil dibatalkan oleh sistem.
<b>Hasil Pengujian</b>	Penghapusan barang jualan berhasil dibatalkan oleh sistem.
<b>Status Pengujian</b>	Valid

### 6.3.23 Pengujian Validasi Melihat Daftar Pesanan

**Tabel 6.58 Kasus uji validasi melihat daftar pesanan**

<b>Kode Kebutuhan</b>	STMK_2300
<b>Nama Kasus Uji</b>	Melihat daftar pesanan
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Toko Komputer menekan tombol “Pesanan”.</li> </ol>



<b>Hasil yang diharapkan</b>	Data daftar pesanan berhasil ditampilkan oleh sistem.
<b>Hasil Pengujian</b>	Data daftar pesanan berhasil ditampilkan oleh sistem.
<b>Status Pengujian</b>	Valid

**Tabel 6.59 Kasus uji validasi alternatif melihat daftar pesanan 2a**

<b>Kode Kebutuhan</b>	STMK_2300
<b>Nama Kasus Uji</b>	Ketika toko komputer tidak melakukan <i>login</i> saat akan melihat daftar pesanan
<b>Prosedur</b>	1. Toko Komputer menekan tombol “Pesanan”.
<b>Hasil yang diharapkan</b>	Halaman <i>login</i> akan ditampilkan oleh sistem
<b>Hasil Pengujian</b>	Halaman <i>login</i> akan ditampilkan oleh sistem
<b>Status Pengujian</b>	Valid

### 6.3.24 Pengujian Validasi Melihat Detail Pesanan

**Tabel 6.60 Kasus uji validasi melihat detail pesanan**

<b>Kode Kebutuhan</b>	STMK_2400
<b>Nama Kasus Uji</b>	Melihat detail pesanan
<b>Prosedur</b>	1. Toko Komputer menekan tombol “Pesanan”. 2. Toko Komputer menekan tombol “Detail”.
<b>Hasil yang diharapkan</b>	Halaman detail pesanan berhasil ditampilkan oleh sistem.
<b>Hasil Pengujian</b>	Halaman detail pesanan berhasil ditampilkan oleh sistem.
<b>Status Pengujian</b>	Valid

**Tabel 6.61 Kasus uji validasi alternatif melihat detail pesanan 1a**

<b>Kode Kebutuhan</b>	STMK_2400
<b>Nama Kasus Uji</b>	Ketika tidak ada pesanan saat akan melihat <i>detail</i> pesanan
<b>Prosedur</b>	1. Toko Komputer menekan tombol “Pesanan”.

<b>Hasil yang diharapkan</b>	Tombol “Detail” tidak akan ditampilkan oleh sistem
<b>Hasil Pengujian</b>	Tombol “Detail” tidak akan ditampilkan oleh sistem
<b>Status Pengujian</b>	Valid

### 6.3.25 Pengujian Validasi Memasukkan Data Pengiriman Barang

**Tabel 6.62 Kasus uji validasi memasukkan data pengiriman barang**

<b>Kode Kebutuhan</b>	STMK_2500
<b>Nama Kasus Uji</b>	Memasukkan data pengiriman barang
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Toko Komputer menekan tombol “Sudah Kirim Barang?”.</li> <li>2. Toko Komputer memasukkan data pada formulir pengiriman barang yang berisi nama kurir dan nomor resi.</li> </ol>
<b>Hasil yang diharapkan</b>	Data pengiriman barang berhasil disimpan oleh sistem.
<b>Hasil Pengujian</b>	Data pengiriman barang berhasil disimpan oleh sistem.
<b>Status Pengujian</b>	Valid

**Tabel 6.63 Kasus uji validasi alternatif memasukkan data pengiriman barang 2a**

<b>Kode Kebutuhan</b>	STMK_2500
<b>Nama Kasus Uji</b>	Ketika salah satu kolom pada formulir pengiriman barang tidak diisi oleh Toko Komputer
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Toko Komputer menekan tombol “Sudah Kirim Barang?”.</li> <li>2. Toko Komputer memasukkan data pada formulir pengiriman barang yang berisi nama kurir dan nomor resi.</li> </ol>
<b>Hasil yang diharapkan</b>	Pesan peringatan yang berisi “harap isi bidang ini” ditampilkan oleh sistem.
<b>Hasil Pengujian</b>	Pesan peringatan yang berisi “harap isi bidang ini” ditampilkan oleh sistem.
<b>Status Pengujian</b>	Valid



Setelah dilakukannya uji kasus pada pengujian validasi dengan menggunakan teknik pengujian *black box* dengan metode *test – criteria* dihasilkan sebagai berikut :

1. Tidak ditemukannya fungsi yang tidak Valid, semua fungsi dapat berjalan dengan baik sesuai dengan kebutuhan yang telah di rancang sebelumnya.
2. Tidak ditemukannya kesalahan antarmuka, semua halaman dapat tampil dan berjalan dengan baik sesuai pada tahapan perancangan sebelumnya.
3. Tidak ditemukannya kesalahan pada akses database, semua data pada database dapat diakses dan diproses oleh sistem.
4. Tidak ditemukannya kesalahan pada perilaku sistem, sistem yang telah diujikan memiliki fungsi yang sesuai dengan kebutuhan pada tahapan perancangan.

Tidak ditemukannya kesalahan Inisialisasi, semua Inisialisasi objek dapat dilakukan sesuai dengan kebutuhan sistem.



## BAB 7 PENUTUP

### 7.1 Kesimpulan

Setelah dilakukannya pengembangan sistem penjualan dengan rekomendasi kompatibilitas komponen pada toko komputer dengan menggunakan pendekatan berorientasi objek. Dihasilkan analisis 25 kebutuhan fungsional sistem dan 3 aktor pada sistem.

Dari hasil OOA dilakukan perancangan kembali dengan *Object Oriented Design* (OOD). OOD melakukan perancangan pemodelan *class* diagram guna mengidentifikasi *class – class* apa saja yang akan ada pada sistem. *Entity Relationship* Diagram (ERD) dibutuhkan untuk menggambarkan hubungan tiap *entity* yang ada pada sistem dan *Physical Data Model*. Hasil dari OOD tersebut diimplementasikan sistem dengan *Object Oriented Programming* (OOP), *framework* Laravel digunakan sebagai *framework* untuk mengembangkan sistem dengan menggunakan bahasa pemrograman PHP dengan bantuan teknologi dalam mempercantik tampilan halaman seperti HTML, Javascript, CSS, dan *Bootstrap*.

Pada pengembangan sistem penjualan dengan rekomendasi kompatibilitas komponen pada toko komputer digunakan 2 metode. Pada pengujian *white box* menguji pada bagian *pseudocode* yang sudah didefinisikan di dalam perancangan komponen dan menghitung *cyclomatic complexity*, setelah itu dilakukan pengujian unit sesuai dengan hasil dari perhitungan *cyclomatic complexity*. Pada pengujian unit tiga *method* yang diuji yaitu *method* *create()*, *method* *simulasi()*, *method* *bayar()* yang menghasilkan status valid dan menghasilkan persentase pengujian *white box* testing yaitu 100%. Pada pengujian integrasi yang menggunakan unit terbesar yang diuji yaitu *method* *bayar()* dikarenakan dalam *method* ini menjalankan 3 fungsi sekaligus yaitu fungsi *cekPhoto*, *order*, *ubahStatus*. Pada *black box testing* dilakukan pengujian dengan membuat *test case* yang sesuai dengan jumlah kebutuhan yang didapatkan dari proses analisis kebutuhan. Hasil dari menguji *xx test case* menyatakan *xx test case* menghasilkan status valid. Pengujian *black box* testing memiliki persentase keberhasilan 100%. Berdasarkan hasil pengujian, dapat disimpulkan bahwa sistem penjualan dengan rekomendasi kompatibilitas komponen pada toko komputer menghasilkan persentase 100%.

### 7.2 Saran

Kebanyakan peminat yang ingin membeli satu set komputer ingin juga melihat komparasi dari setiap hasil daftar komponen komputer yang berada di halaman simulasi. Hal ini bertujuan untuk membandingkan harga total dari daftar komponen komputer yang satu dengan yang lain, agar peminat tahu perbedaan harga dan juga komponen komputer yang telah dipilih.



Pada bagian simulasi dapat dibuatkan halaman formulir menjadi responsive dengan cara menampilkan formulir berdasarkan kapasitas motherboard. Sebagai contoh, bila motherboard yang telah di pilih oleh dapat mengakomodir 6 Storage dengan interface SATA maka secara otomatis membuat 6 table khusus untuk pengguna memilih storage yang tipe interfacenya adalah SATA.

Untuk pembangunan sistem berikutnya, perlu dibuatnya sebuah sistem penjualan yang tidak hanya menampilkan komponen komputer dari satu toko saja, melainkan juga menampilkan lebih dari satu toko. Sehingga sistem dapat mengakomodir lebih dari satu toko, agar dapat memberikan opsi tambahan kepada pengguna.



## DAFTAR REFERENSI

- Anhar. 2010. *Panduan Menguasai PHP dan MySQL Secara Otodidak*. Jakarta: PT TransMedia.
- Britton, C., & Doake, J. 2005. *A Student Guide to Object-Oriented Development*. Inggris: Elsevier Butterworth-heinemann.
- Chacos, B. 2013. *How to Build PC*. How to avoid common PC building mistakes. Tersedia di: <https://www.pcworld.com/article/2049100/components/how-to-avoid-common-pc-building-mistakes.html> [Diakses 31 Juli 2019]
- Edmons, Rich. 2018. Working Together. *How to Make Sure All Your PC Components Are Compatible*. Tersedia di: <https://www.windowscentral.com/how-make-sure-all-your-pc-components-are-compatible> [Diakses 31 Juli 2018]
- Fauzi, Mohammad. 2019. *Komputer dan Internet*. Bangkes: Duta Media Publishing
- Huda, M. B. K. 2010. *Membuat Aplikasi Database dengan Java, MySQL dan Netbeans*. Jakarta: Elex Media Komputindo.
- Juntak Team Works. 2009. *Panduan Menginstall Komputer; Mudah dan Praktis*. Yogyakarta: MediaKom.
- Lim, T. D. D. 2017. *Analisis dan Perancangan Sistem Informasi Berbasis Online Shop Pada Toko Fashion Era*. Batam: Sekolah Tinggi Manajemen Informatika dan Komputer GICI Batam.
- Marsic, Ivan. 2012. *Software Engineering*. New Jersey : Rutgers University.
- McCool, S. 2012. *Laravel Starter*. United Kingdom: Packet Publishing Ltd.
- Mufadhol. 2017. *Buku Ajar Arsitektur Sistem Komputer*. Semarang: Sekolah Tinggi Elektronika & Komputer.
- Mujiyana, I. E. 2013. *Analisis Faktor – Faktor yang Mempengaruhi Keputusan Pembelian via Internet pada Toko Online*. Semarang: Universitas Diponegoro.
- Niswara, B. A. H. T., Putri, R. R. M., & Hidayat, N. 2018. *Rekomendasi Pemilihan Paket Personal Computer Menggunakan Metode AHP-TOPSIS*. Malang: Universitas Brawijaya.
- Peranginangin, K. 2006. *Aplikasi WEB dengan PHP dan MySQL*. Yogyakarta: Andi.
- Pressman, R. S. 2010. *Software Engineering: A Practitioner's Approach*. New York: McGraw-Hill.
- Priyanto, Duwi. 2012. *Panduan Lengkap Komputer: PC, Laptop, Tabler*. Depok: Puspa Swara.



Sarwandi. 2016. *Toko Online Modern dengan Opencart*. Jakarta: PT Elex Media Komputindo.

Satzinger, J. W., Jackson, R. B., & Burd, R. D. 2010. *System Analyst And Design in A Changing World (7th Edition)*. Boston: Cengage Learning.

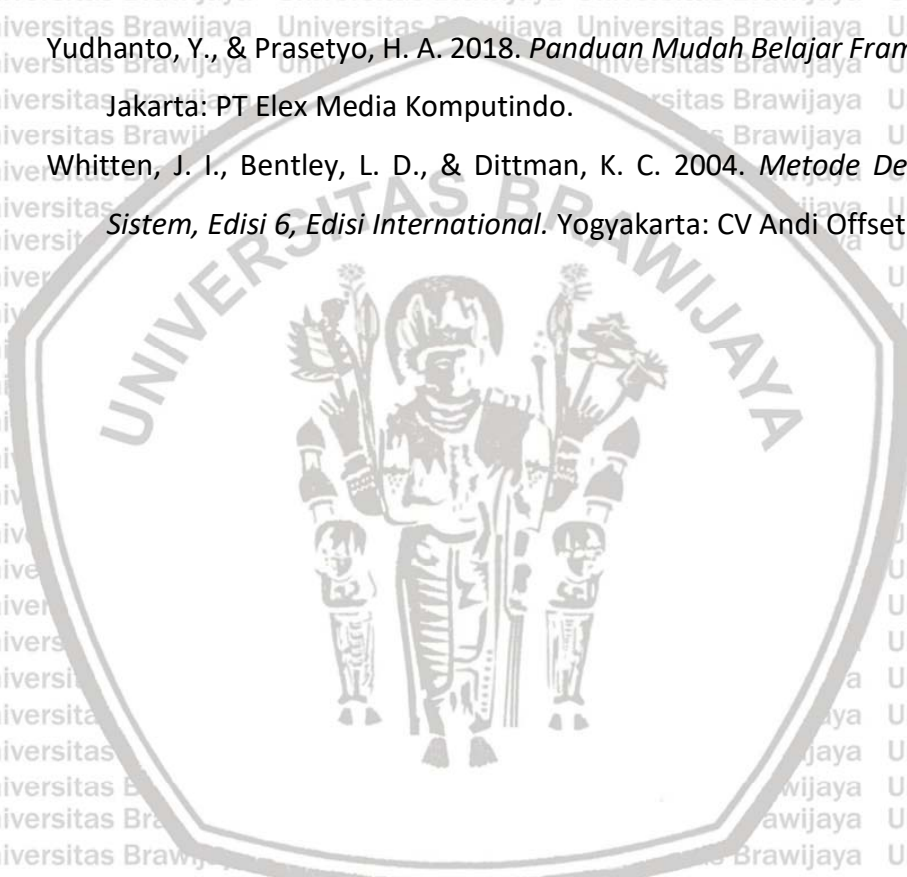
Sommerville, I. 2011. *Software Engineering (9th Edition)*. Pearson.

Sutono. (2014). *Laboratorium Hardware Universitas Komputer Indonesia*. Bandung: Universitas Komputer Indonesia.

Yani, Ahmad. 2005. *Panduan Menjadi Teknisi Komputer*. Jakarta: PT Kawan Pustaka.

Yudhanto, Y., & Prasetyo, H. A. 2018. *Panduan Mudah Belajar Framework Laravel*. Jakarta: PT Elex Media Komputindo.

Whitten, J. I., Bentley, L. D., & Dittman, K. C. 2004. *Metode Desain & Analisis Sistem, Edisi 6, Edisi International*. Yogyakarta: CV Andi Offset.



## LAMPIRAN

Dalam Penelitian ini, penulis melakukan wawancara terhadap narasumber yang memberikan kontribusi dan informasi dalam penelitian ini. Beliau adalah Bapak Faturrohman sebagai Marketing Manager dari toko komputer yang berada di Malang yang bernama KiosKomputer.

### 1. Pertanyaan wawancara untuk narasumber:

- a. Apakah saya diperbolehkan melakukan wawancara di toko Anda mengenai dunia perakitan komputer?
- b. Bagaimana proses yang dilewati oleh pembeli ketika pembeli ingin membeli sebuah barang langsung dari toko?
- c. Apakah masih ada kasus dimana pembeli mengembalikan barang yang telah dibeli dengan alasan barang tersebut ternyata tidak bisa digunakan dengan komputer yang ada di rumah?
- d. Dalam kurun waktu satu bulan berapa kali kejadian pengembalian barang dapat terjadi?
- e. Apakah ada kerugian yang dialami oleh toko pada saat pembeli melakukan pengembalian barang?
- f. Apakah pada saat proses pemilihan barang, pembeli tidak menanyakan detail dan kondisi barangnya?
- g. Berapa lama proses tanya menanya pembeli sebelum pembeli memutuskan untuk membeli barang?

### 2. Hasil Wawancara dengan narasumber:

- a. Iya boleh, mas. Silahkan, monggo.
- b. Biasanya orang datang ke toko terus langsung bilang budgetnya berapa. Setelah itu kita bikin daftar komponennya. Lalu, menunjukan hasilnya ke calon pembeli. Ada juga yang datang ke toko sudah membawa daftar komponen yang ingin dibeli, terus kita cek apakah barang yang ada di komponen itu oke apa tidak. Kalau oke kita ambilkan barang. Ada juga yang bertanya kalo buat ngame dengan budget sekian bagusnyanya bagaimana.
- c. Kalo itu, masih banyak orang masih awam tentang masalah komputer apa itu ddr? Apa itu storage? Apa itu SSD? Masih banyak orang yang nggak paham dengan masalah itu dan juga banyak orang yang kembali untuk mengembalikan barangnya karena salah beli. Itu paling banyak. Soalnya mereka belum paham yang mau dibeli itu apa aja dia itu belum paham.
- d. Nggak terlalu sering sih mas, mungkin sekitar 2-3 kali aja karena sebelum mereka beli kita sebagai toko nanya dan ngasih tau dulu.



e. Ada tentunya karena barang yang sudah dikembalikan kami jual harganya sedikit lebih rendah karena sudah dalam kondisi terbuka di luarbox. Walaupun kejadian seperti itu tidak sering terjadi.

f. Orang yang mau beli seperti itu banyak, cuma kita kasih saran langsung daripada processor seri untuk overclock namun motherboard belum seri overclock kita kasih saran untuk processor yang belum di-overlock, soalnya nanti percuma jadinya tidak akan berfungsi nantinya.

### 3. Dokumentasi hasil wawancara :

